DOCUMENT RESUME

ED 054 704	48	FL 002 629

AUTHOR	Price, James D.
TITLE	A Computerized Phrase-Structure Grammar (Modern Hebrew): Part III, An Algorithm for Generating Hebrew Sentences. Final Report.
INSTITUTION	Franklin Inst. Research Labs., Philadelphia, Pa.
SPONS AGENCY	Office of Education (DHEW), Washington, D.C. Bureau of Research.
BUREAU NO	BR-9-7722
PUB DATE	Jun 71
CONTRACT	OEC-0-9-097722-4411
NOTE	318p.
EDRS PRICE	MF-$0.65 HC-$13.16
DESCRIPTORS	*Algorithms; Computational Linguistics; *Computer Programs; Deep Structure; Flow Charts; Grammar; *Hebrew; Language Patterns; Language Research; Language Universals; Logic; *Phrase Structure; Semitic Languages; Sentence Structure; Syntax; Tables (Data); *Transformation Generative Grammar; Transformation Theory (Language)

ABSTRACT

The third part of the four-part report of research on the development of a computerized, phrase-structure grammar of modern Hebrew discusses a computerized algorithm for generating sentences in modern Hebrew based on a generalized complex-constituent-phrase structure grammar as it is applied to the syntax and orthography of the language as described in Part II. After a discussion of the structural model, a description of the algorithm is provided along with a description of the computer program and the procedures for generating sentences. The algorithm is demonstrated and tested. Sentences generated are given along with the grammar rules which are tested. Appendix A lists the grammar rules for modern Hebrew syntax for the synthesis of sentences. Appendix B contains the source-language listing of the computer program and associated subprograms. Appendix C provides a sample output of the subprogram output. Appendix D gives the list of sentence-description data card types. The algorithm defines all the procedures and associated computer programs for generating sentences in modern Hebrew, but improvements can be made in the grammar upon which it operates. Since it depends only on the form of grammar and not on content, the algorithm is not necessarily limited to the Hebrew language. For related reports see FL 002 627, FL 002 628, and FL 002 630. (Author/VM)

OE/115

Title VI, NDEA, Section 602 PA-48 Bureau#9-772

FL

ED054704

FINAL REPORT
Project No. 097722
Contract No. OEC-0-9-097722-4411
Franklin Institute Report No. F-C2585-3

A COMPUTERIZED PHRASE-STRUCTURE GRAMMAR (MODERN HEBREW)
PART III

James D. Price
Franklin Institute Research Laboratories
20th and Race Street
Philadelphia, Penna. 19103

June 1971

U.S. DEPARTMENT OF HEALTH, EDUCATION & WELFARE
OFFICE OF EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY.

U.S. DEPARTMENT OF
HEALTH, EDUCATION, AND WELFARE
Office of Education
Institute of International Studies

FL002629

ED054704

FINAL REPORT
Project No. 097722
Contract No. OEC-0-9-097722-4411
Franklin Institute Report No. F-C2585-3

A COMPUTERIZED PHRASE-STRUCTURE GRAMMAR (MODERN HEBREW)
PART III

AN ALGORITHM FOR GENERATING HEBREW SENTENCES

James D. Price
Franklin Institute Research Laboratories
Philadelphia, Pennsylvania 19103

June 1971

The research reported herein was performed pursuant to a contract with the Office of Education, U. S. Department of Health, Education, Welfare.

U. S. DEPARTMENT OF
HEALTH, EDUCATION, AND WELFARE
Office of Education
Institute of International Studies

ABSTRACT

This is the third part of a four-part report of research for the development of a Computerized Phrase-Structure Grammar of Modern Hebrew. This part describes a computerized algorithm for generating sentences in modern Hebrew which is based on a generalized complex-constituent phrase-structure grammar (defined in Part I) as it was applied to the syntax and orthography of modern Hebrew (described in Part II). A computer program of the algorithm is described which includes, for the main program and each subprogram, (1) a flow chart, (2) a written description of its operation, and (3) a source language listing in FORTRAN IV.

The algorithm was made operational in a UNIVAC 1108 computer and used to systematically test the grammar of modern Hebrew syntax by generating sentences in the language. A total of 47 sentences were generated in the process of which 113 of the 179 grammar rules were tested. The tests fully demonstrate the capability of the algorithm for generating sentences. Errors which occurred were due to the content of the grammar rules and not to the logic of the algorithm. It proved to be a valuable tool for testing, validating, and, in numerous cases, correcting the grammar rules.

Two aspects of the algorithm need further attention: (1) several computations should be freed from dependence on the "content" of the source grammar, and (2) the input data preparation should be simplified. With these modifications the algorithm can be used to generate sentences in other Semitic languages whenever grammars become available, and to aid in the development of such grammars.

3

TABLE OF CONTENTS

Page

ABSTRACT i

PART III. AN ALGORITHM FOR GENERATING HEBREW SENTENCES. 3-1

3.1 A Structural Model of Hebrew Sentences 3-1

3.1.1 The Mapping Functions 3-2
3.1.2 The Variables 3-6

3.2 The Procedure. 3-9
3.3 Computer Program of the Algorithm. 3-14

3.3.1 Flow-Chart Description of Computer Program. . . 3-14
3.3.2 Input Map for Computer Program. 3-77

3.4 Test and Demonstration of Algorithm. 3-88

3.4.1 Sentences Generated by Algorithm. 3-89
3.4.2 Grammar Rules Tested by the Algorithm 3-93

3.5 Conclusion 3-98

4

PART III

AN ALGORITHM FOR GENERATING HEBREW SENTENCES

This part of the report describes a computerized algorithm for generating Hebrew sentences, and algorithm essentially complete in that it defines all the procedures and associated computer programs for generating sentences in modern Hebrew, but incomplete in that improvements can be made in the grammar upon which it operates. Because the algorithm is dependent only on the "form" of the grammar and not on its "content"[1], it may be used for training research workers in the field of computational linguistics without being limited to the Hebrew language.

The algorithm consists of: (1) a structural model of the language, (2) a procedure for using the model to generate sentences in the language, and (3) a program for performing the procedure by means of a computer.

The first section discusses the structural model of modern Hebrew. The second section provides a detailed description of the procedure for generating sentences. The third section provides a detailed description of the computer program of the procedure. The last section describes the tests and verifications of the algorithm.

3.1 A Structural Model of Hebrew Sentences

The structural model of modern Hebrew sentences consists of a complex-constituent phrase-structure grammar, the generalized characteristics of which are defined in Part I of this report and the specific details for which are defined in Part II. Those details of the structural model required to explain the algorithm are presented in the appropriate places throughout the text. Reference should be made to Parts I and II for further information.

For the purposes of the algorithm, the structural model is viewed as the following:

(1) a set of mapping functions

(2) a set of variables

---

[1] A few exceptions are noted throughout the text; these will be corrected in a subsequent revision.

3.1.1 The Mapping Functions

The mapping functions enable the user to define the specific details of the structural model that are peculiar to the given language (in this case Hebrew). The use of mapping functions makes the algorithm independent of the "content" of the grammar it processes by enabling the user to define the "content" for a given language.

The mapping functions consist of the following:

(1) a table of the transliteration

(2) a table of symbols

(3) a table of grammar rules

(4) a table of restraints

3.1.1.1 The Transliteration

The table of transliteration provides a list of the alphabetic and numeric characters of the given language. These data are used by the algorithm to transform the input characters to numbers for use in computation and to transform the resultant output numbers to their corresponding characters. Table 3-1 is a listing of the transliteration used for Hebrew (see Table 2-1, Part II, for Hebrew equivalents).

3.1.1.2 The Symbols

The table of symbols provides a list of the symbols used in the grammar of the given language. The list is used by the algorithm to transform the input symbol names of the grammar rules to numbers for use in computation and to transform the resultant output numbers to their corresponding symbol names for use in tree diagrams and output listings. Table 3-2 is a listing the symbols used for Hebrew.[2]

3.1.1.3 The Rules

The table of grammar rules provides a list of the rules of the complex-constituent phrase-structure grammar of the given

---

[2] See Table 2-2 and 2-3 of Part II for further description of the symbols.

6

Table 3-1

ALPHA NUMERIC TRANSLITERATION

L	TRANSL(L)	IA	L	TRANSL(L)	IA
1	0	0	26	O	25
2	1	1	27	P	26
3	2	2	28	&	27
4	3	3	29	Q	28
5	4	4	30	R	29
6	5	5	31	S	30
7	6	6	32	T	31
8	7	7	33	(	32
9	8	8	34	)	33
10	9	9	35	,	34
11	A	10	36	:	35
12	B	11	37	?	36
13	D	12	38	.	37
14	G	13	39	!	38
15	H	14	40	–	39
16	W	15	41	(space)	0
17	Z	16	42	F	41
18	X	17	43	I	42
19	@	18	44	V	43
20	Y	19	45	*	44
21	K	20	46	#	45
22	L	21	47	Δ	46
23	M	22	48	'	47
24	N	23	49	?	48
25	C	24	50	?	49

Table 3-2

LIST OF SYMBOLS

SYMBOL No.	SYMBOL Name	SYMBOL No.	SYMBOL Name	SYMBOL No.	SYMBOL Name
1	Z	21	B_p	41	R_{sp}
2	V_{qo}	22	N_{pb}	42	N_{sp}
3	A_{pa}	23	N_{pa}	43	N_{op}
4	A_p	24	N_{pc}	44	N_{ip}
5	A_s	25	N_{ap}	45	N_{px}
6	N_a	26	N_p	46	V_{ma}
7	S_{qo}	27	D_{pa}	47	V_{mb}
8	N_s	28	D_{pb}	48	V_{mc}
9	R_d	29	D_{pc}	49	V_{md}
10	R_o	30	D_p	50	V_m
11	B_{aa}	31	E_a	51	V_{mr}
12	B_{ab}	32	V_b	52	V_{mi}
13	B_{ac}	33	V_{bb}	53	V_p
14	B_{ad}	34	V_c	54	V_{rb}
15	B_{ae}	35	V_{aa}	55	V_{ri}
16	B_{af}	36	V_a	56	N_v
17	B_{ba}	37	B_c	57	N_w
18	B_{bb}	38	X_p	58	E_{pb}
19	B_a	39	N_o	59	E_{pa}
20	B_{bc}	40	D_{pd}	60	E_p

Table 3-2 (Continued)

LIST OF SYMBOLS

SYMBOL No.	SYMBOL Name	SYMBOL No.	SYMBOL Name
61	S_{aa}	81	E
62	S_{ab}	82	G
63	S_{ac}	83	H
64	S_{a}	84	I
65	S_{ro}	85	J
66	S_{ri}	86	L
67	R_{g}	87	N
68	K_{n}	88	O
69	K_{c}	89	P
70	K_{k}	90	Q
71	K_{i}	91	R
72	K_{d}	92	T
73	S_{d}	93	U
74	S	94	V
75	S_{i}	95	W
76	S_{c}	96	Y
77	A	97	F
78	B		
79	C		
80	D		

language. Appendix A of this part of the report contains a listing of the grammar rules used for Hebrew.[3]

3.1.1.4 The Restraints

The table of restraints provides a list of limitations to be used by the rules of the grammar. The grammar specifies a certain horizontal row of the table to limit the value of a given symbol subscript. The restraints are interpreted as follows:

(1) The first number in the row specifies the number of restraint values in the row.

(2) If the numbers in the specified row are positive, the value of the given subscript must be one of the numbers in the row.

(3) If the numbers in the specified row are negative, the value of the given subscript must not be one of the numbers in the row.

Table 3-3 is a list of the restraints used in the grammar of Hebrew.

3.1.2 The Variables

The set of variables consist of 29 subscripts on the set of symbols of the grammar that enable the user to define the sentence to be generated by the algorithm. The algorithm uses three types of variables:

(a) Fixed variables -- those with values fixed by the rules of the grammar.

(b) Dependent variables -- those the values of which are computed by the grammar.

(c) Independent variables -- those the value of which the user defines in the process of describing the sentence to be generated.

Table 3-4 is a list of the variables used by the algorithm together with the description of each.[4]

---

[3] Reference should be made to Section 2.2.3 of Part II of this report for a detailed description of the rules.

[4] See Section 2.2.1 of Part II for a detailed description of the variables as they apply to Hebrew.

Table 3-3

LIST OF RESTRAINTS

Row No.	No. of Items	Item			
		1	2	3	4
1	1	-1			
2	1	-2			
3	1	-3			
4	2	0	1		
5	2	1	2		
6	2	-1	-2		
7	2	1	3		
8	2	2	5		
9	2	4	5		
10	2	6	7		
11	3	1	2	3	
12	3	4	5	6	
13	4	1	2	3	4
14	4	-1	-2	-3	-4

Table 3-4

LIST OF VARIABLES

Var. No.	Var. Name	Description	Var. No.	Var. Name	Description
1	SN	Symbol No.	16	i	mood
2	m	opt./mand.	17	t	tense
3	f	comp. class	18	s	stem
4	k	comp. no.	19	w_1	Root 1
5	b	comp. type	20	w_2	Root 2
6	c	class	21	w_3	Root 3
7	ℓ	neg. class	22	w_4	Root 4
8	y	neg./pos	23	ST	Symbol Type
9	d	definiteness	24	RN	Rule No.
10	n	number	25	EN	Element No.
11	g	gender	26	NRH	No. Rt. Elements
12	p	person	27	RT	Restraint type
13	r	prep. class	28	RS	Restraint Subs.
14	a	verb. class	29	x	No./gend. Trans.
15	v	voice			

3.2 The Procedure

This section describes the procedure for using the structural model to generate Hebrew sentences. It consists of the logical interaction of a set of operational functions that manipulate the data of the mapping function in accordance with the specified values of the independent variables supplied by the user. Basically the procedure begins with an initial symbol of the Grammar of Hebrew Snytax[5] and applies the replacement rules of the grammar until only terminal symbols remain. The major operational functions required to perform this task are listed below:

(1) Read in the initial symbol

(2) Write the initial symbol on the "A" list of symbols

(3) Initialize symbol counting indices

(4) Increment symbol counting index

(5) Compare symbol counting index with the computed maximum value

(6) Read a symbol from the "A" list of symbols

(7) Determine whether symbol is terminal

(8) Write a symbol on the "B" list

(9) Locate a grammar rule that applies to a given symbol

(10) Determine that the rule matches the given symbol with respect to all subscripts

(11) Compute values for dependent-variable subscripts

(12) Initialize rule element counting indices

(13) Increment rule element counting index

(14) Compare rule element counting index with the computed maximum value

(15) Define values of independent-variable subscripts by reading input

(16) Omit non-mandatory symbols

---

[5] Defined in Part II of this report.

(17) Write mandatory rule element on the "B" list.

(18) Determine whether all symbols on the "B" list are terminal

(19) Erase the "A" list of symbols and transfer the symbols on the "B" list to the "A" list

(20) Compute the orthography of terminal symbols

(21) Construct a tree diagram of the generated sentence.

Many minor operational functions that are associated with these are not listed. They are defined in Section 3.3 that describes the computer program of the algorithm. The above are sufficient for explaining the procedure of the algorithm.

The procedure consists of a logical manipulation of the operational functions so as to generate a sentence. The logical interrelationship of the functions is defined in flow chart form in Figures 3.1a and 3.1b.

The initial symbol is read in (Block 1)[6] together with the value of all independent variable subscripts, and it is written on the "A" list of symbols (LIST1) as the first symbol (Block 2). The number of symbols on LIST1 (J_{max}) is computed and counting register (J) is initialized (Block 3). The counting register (J) is indexed by one (Block 4) and the value of J is compared with J_{max} to determine whether or not all the symbols on list LIST1 have been processed (Block 5); if not, computation procedes to Block 6, otherwise to Block 18.

In Block 6, the J-th symbol of LIST1 is read. If it is a terminal symbol (Block 7) it is written onto the "B" list (LIST2) with no changes (Block 8) and computation returns to Block 4. If it is not a terminal symbol (Block 7), the first applicable grammar rule is located (Block 9) and tested (Block 10) to determine that the left hand element of the rule matches the J-th symbol of LIST1 in accordance with the following criteria:

(1) the symbol number must be the same for both symbols, or if not, the grammar rule symbol must be a variable symbol; and

(2) the symbol class must be the same for both symbols; and

---

[6] The "Block" number of the figure corresponds to operation function number previously listed.

61

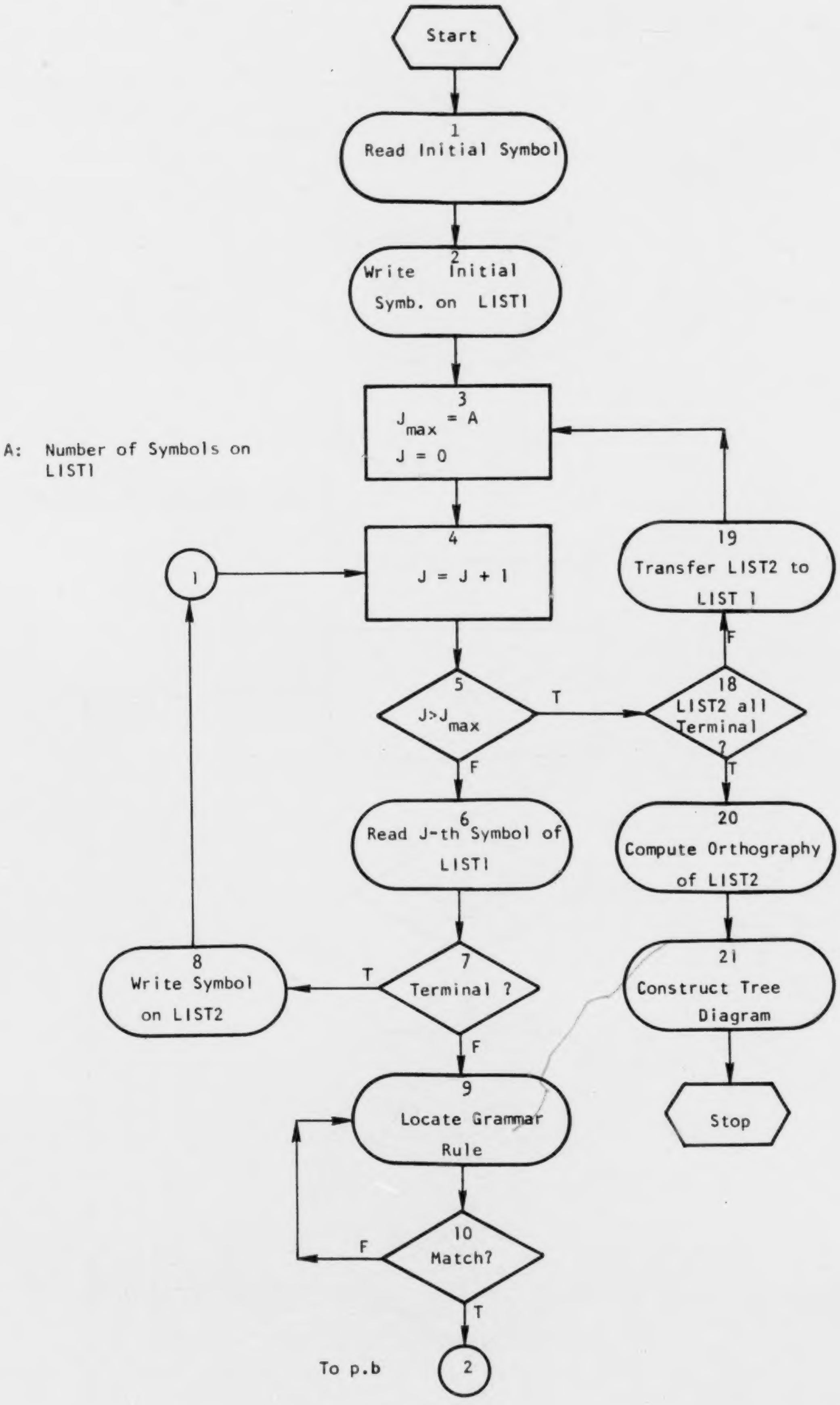

Start
1
Read Initial Symbol
2
Write Initial
Symb. on LIST1
3
J_{max} = A
J = 0
A: Number of Symbols on LIST1
4
J = J + 1
1
5
J>J_{max}
T
F
6
Read J-th Symbol of LIST1
7
Terminal ?
T
F
8
Write Symbol on LIST2
9
Locate Grammar Rule
10
Match?
F
T
To p.b
2
18
LIST2 all Terminal ?
F
T
19
Transfer LIST2 to LIST 1
20
Compute Orthography of LIST2
21
Construct Tree Diagram
Stop

Figure 3.1a Flow Chart of Algorithm

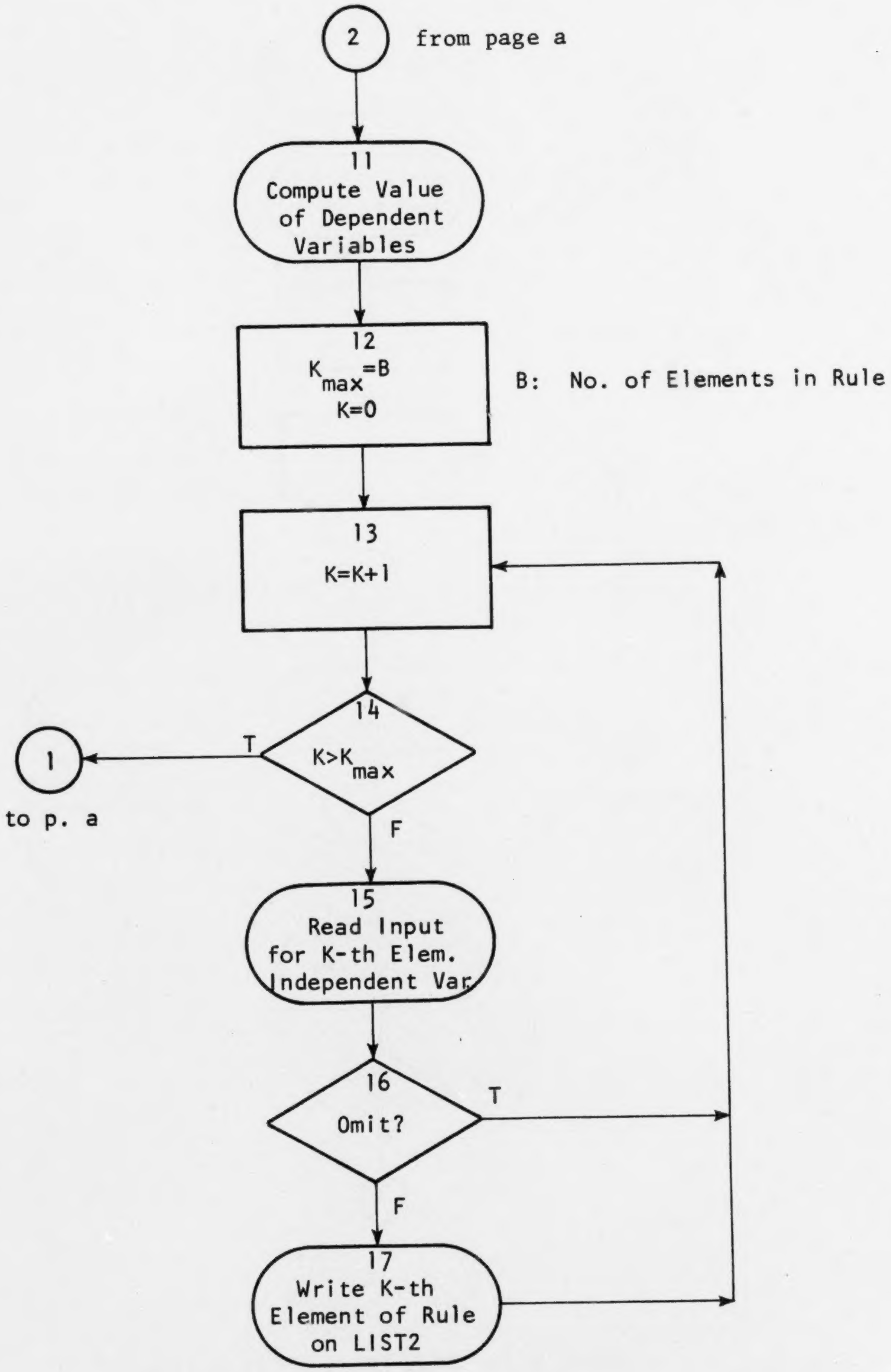

2
from page a
11
Compute Value of Dependent Variables
12
$K_{max}=B$
K=0
B: No. of Elements in Rule
13
K=K+1
14
$K>K_{max}$
T
F
1
to p. a
15
Read Input for K-th Elem. Independent Var.
16
Omit?
T
F
17
Write K-th Element of Rule on LIST2

Figure 3.1b Flow Chart of Algorithm

(3) every fixed-valued subscript of the rule symbol must be the same as the corresponding subscript of the J-th symbol; and

(4) the restraints on the rule symbol must be met.

If the criteria are not met (Block 10) computation returns to Block 9 for location of the next applicable grammar rule. If the criteria are met, the values of the dependent variable subscripts of the given grammar rule are computed (Block 11). This is accomplished by assigning the value of the corresponding subscript of the J-th symbol to the given dependent variable subscripts of the rule.

Next, the number of symbols on the right side of the rule (K_{max}) is computed and counting register (K) is initialized (Block 12) in preparation for writing the K_{max} elements of the right side of the rule onto LIST2. Then register K is indexed by one (Block 13) and the value of K is compared with K_{max} (Block 14) to determine whether or not all the right hand elements of the rule have been processed; if not, computation procedes to Block 15, otherwise to Block 4 for rewriting the next symbol of LIST1. In Block 15, the values of the independent variable subscripts of the k-th element of the rule are read in. The values of these variables are assigned by the user as part of the input data defining the sentence being generated.

The symbols of the grammar have a subscript m that defines the optional/mandatory attribute of the symbol. Most symbols are mandatory (m = 1), but a few are optional (m = 9). For optional symbols, subscript m is an independent variable the value of which is assigned in Block 15.

In Block 16 the assigned value of subscript m is examined. If the symbol is to be omitted (m = 0) computations is returned to Block 13 for processing the next element of the rule. If the symbol is not to be omitted (m = 1), the K-th element of the rule is written on LIST2 (Block 17) with the computed values assigned to its subscripts, and computation returns to Block 13 for processing the next element of the rule.

Referring back to Block 5, if all symbols on LIST1 have been processed, computation procedes to Block 18 where a test is made to determine whether all symbols written on LIST2 were terminal symbols. If so, the desired sentence has been generated and computation precedes to Block 20, if not, the symbols on LIST1 are erased, those on LIST2 are transferred to LIST1, LIST2 is erased, and computation returns to Block 3 for processing the new symbols on LIST1.

Referring back to Block 18, if all symbols on LIST2 were terminal symbols, the orthography of each terminal symbol is computed and edited (Block 20), a tree diagram of the generated sentence is constructed (Block 21), and computation stops.

3.3 Computer Program of the Algorithm

This section describes a computer program of the Algorithm for Generating Hebrew Sentences. First a flow-chart description of the program is given, then an input map is provided with instructions on how to use the program. Appendix B of this part of the report contains a source language listing of the program in FORTRAN IV.

3.3.1 Flow-Chart Description of Computer Program

This section describes the computer program of the Algorithm for Generating Hebrew Sentences in terms of flow diagrams. The program consists of a main program SENSYN and the following subprograms:

(1) ALPHA

(2) ALPHA1

(3) CONSEN

(4) DIAGRM

(5) INPUT

(6) LIMIT

(7) ORTHOG

(8) OUTPUT

(9) RERITE

(10) RULENO

(11) RULRIT

(12) SYMACH

(13) VARATT

(14) INFLEX

Figure 3.2 is a map of the program showing the hierarchy of the calling sequences. The following sections contain descriptions of each portion of the program.

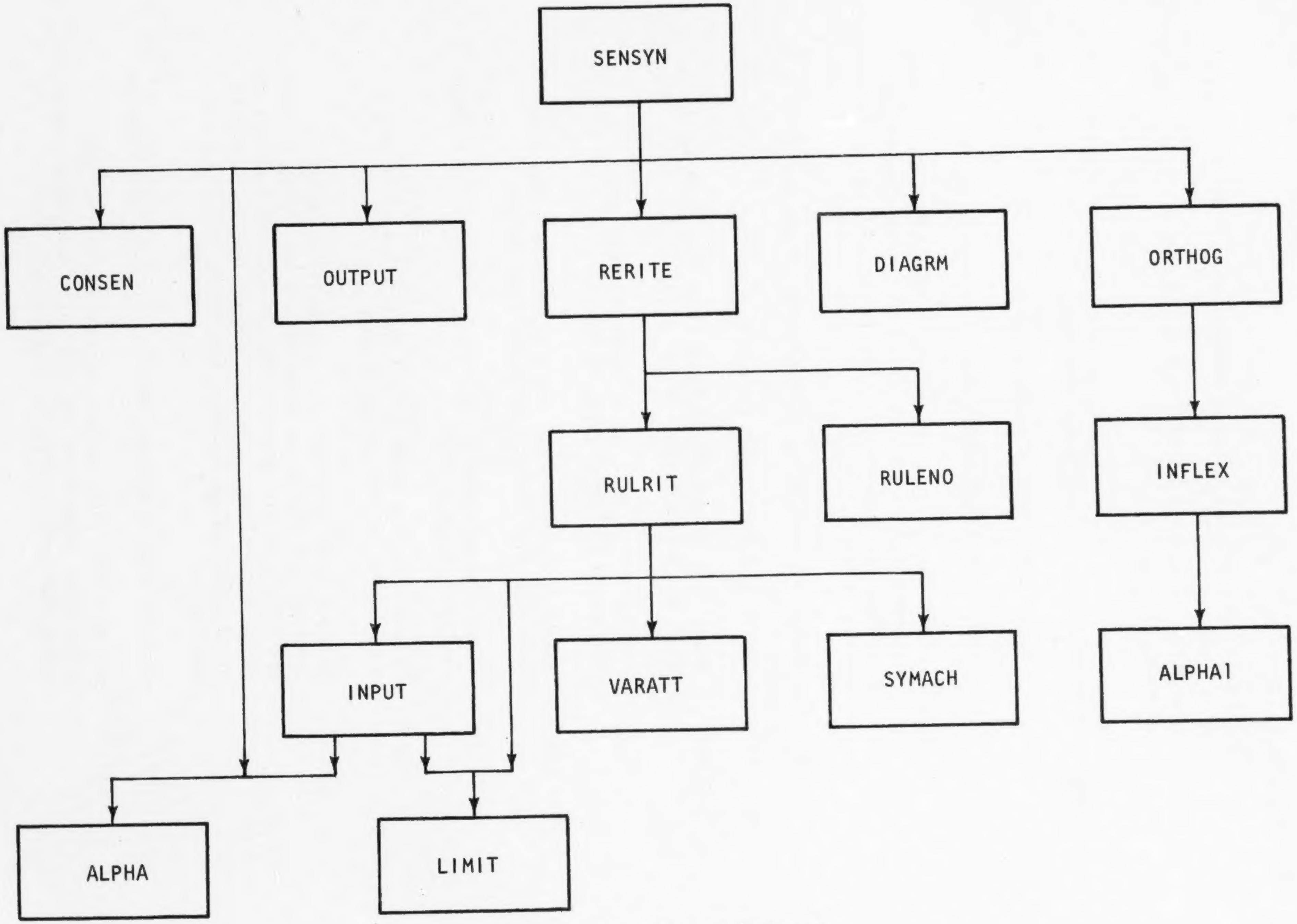

SENSYN
CONSEN
OUTPUT
RERITE
DIAGRM
ORTHOG
RULRIT
RULENO
INFLEX
INPUT
VARATT
SYMACH
ALPHA1
ALPHA
LIMIT

Figure 3.2 Map of Program

3.3.1.1 Main Program SENSYN

This section describes the operation of the main program of the Algorithm for Generating Hebrew Sentences. This program manages the overall operation of the algorithm and calls the various subprograms at the appropriate times.

Figures 3.3 is a flow diagram of main program SENSYM. The main program performs the following operations:

(1) Reads the program options (Fig. 3.3a)

(2) Reads the Transliteration Table (Fig. 3.3a)

(3) Reads the Matrix of Grammar Rules, transforms the alpha-numeric data to integers using Subprogram ALPHA, and stores the transformed rules in Matrix RULE (Fig. 3.3b).

(4) Computes a catalog of the Rules (Fig. 3.3b)

(5) Reads the Restraint Table (Fig. 3.3c)

(6) Reads the table of Symbols that are associated with a symbol number (Fig. 3.3c)

(7) Writes these data on a magnetic tape for permanent storage (Fig. 3.3c)

(8) Reads the equivalent English sentence and the data for the initial symbol (Fig. 3.3c)

(9) Stops the program after the analysis of the last sentence (Fig. 3.3d)

(10) Prints out the data on the initial symbol using Subprogram OUTPUT (Fig. 3.3.d)

(11) Makes a pass through the present string of symbols and replaces each symbol by the symbol(s) specified by the grammar rules and the associated input data (Fig. 3.3d). This operation consists of the following 3 steps for each symbol in the string:

(12) Compute the constants related to compound symbols

(13) Replace the I-th symbol of the string by the symbols(s) specified by the rules and input data using Subprogram RERITE (Fig. 3.3e)

(14) Compute node governing data for use in constructing a tree diagram (Fig. 3.3e)

(15) Prints out the new string of symbols using Subprogram OUTPUT (Fig. 3.3e)

(16) Tests to see if all symbols in the present string are terminal symbols, if not, repeat from step (11), otherwise, continue on (Fig. 3.3f)

(17) Computes node governing data for the last pass on the string for use in constructing a tree diagram (Fig. 3.3f)

(18) Applies the context-sensitive rules to the last string of symbols using Subprogram CONSEN (Fig. 3.3f)

(19) Constructs a tree diagram of the generated sentence using Subprogram DIAGRM (Fig. 3.3f)

(20) Computes the correct orthography of each word edits and prints out the generated Hebrew sentence using Subprogram ORTHOG (Fig. 3.3f)

(21) Returns to Step (8) for next sentence to be generated.

MAIN PROGRAM SENSYN

Note 1:

MODE: Defines the source of the grammar rules and associated data.

MODE=0, data is on magnetic tape ITAPE.

MODE=1, data is to be read from cards.

ITAPE: Defines the logical unit of the computer on which the magnetic tape containing the grammar rules and associated data is mounted.

ITRACE: Enables the user to request the printing of diagnostic trace messages which are printed each time certain subroutines are called.
ITRACE≠0 for trace messages.

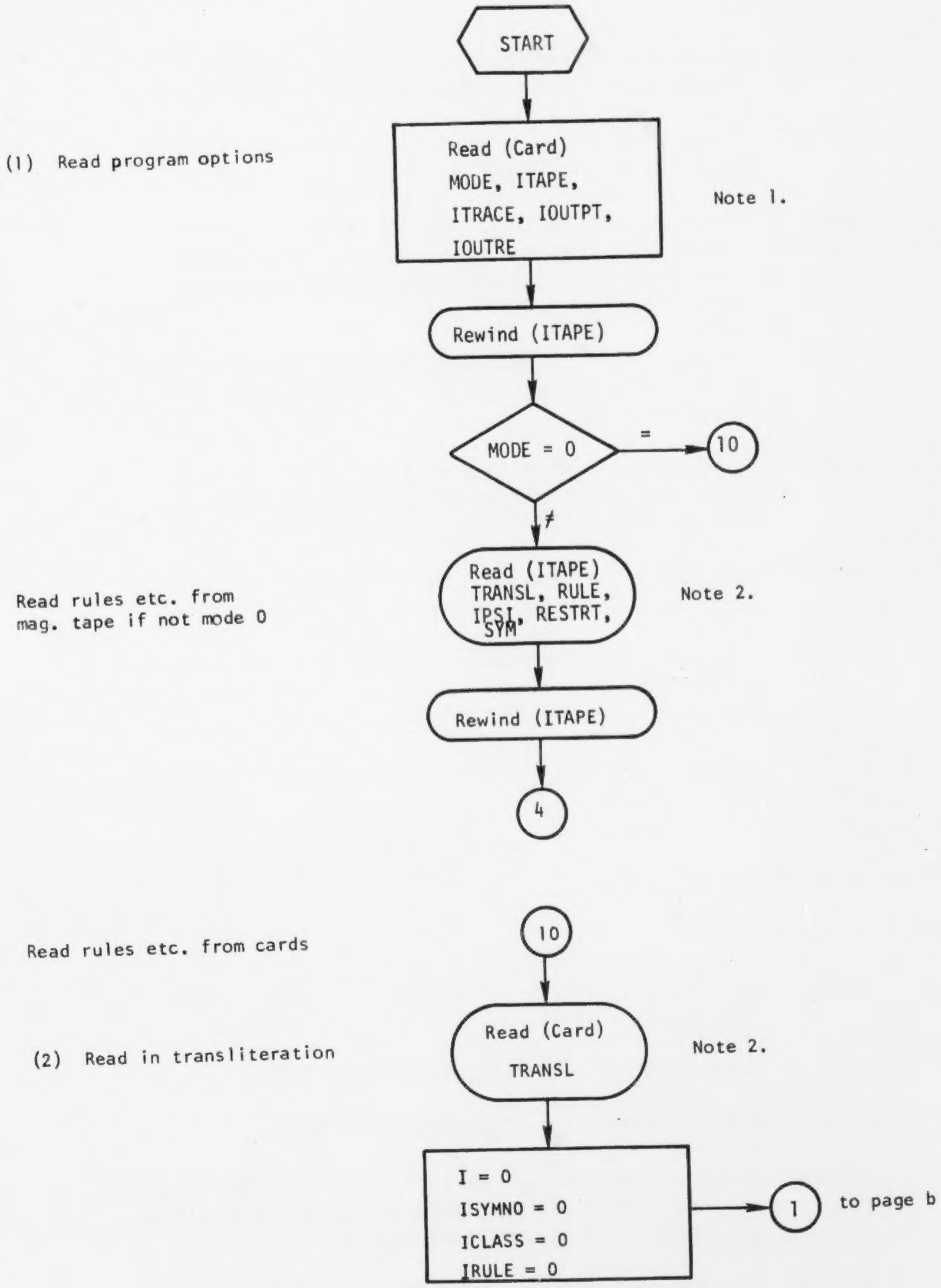

START
(1) Read program options
Read (Card)
MODE, ITAPE,
ITRACE, IOUTPT,
IOUTRE
Note 1.
Rewind (ITAPE)
MODE = 0
=
10
≠
Read rules etc. from
mag. tape if not mode 0
Read (ITAPE)
TRANSL, RULE,
IPSI, RESTRT,
SYM
Note 2.
Rewind (ITAPE)
4
Read rules etc. from cards
10
(2) Read in transliteration
Read (Card)
TRANSL
Note 2.
I = 0
ISYMNO = 0
ICLASS = 0
IRULE = 0
1
to page b

Figure 3.3a Main Program SENSYN

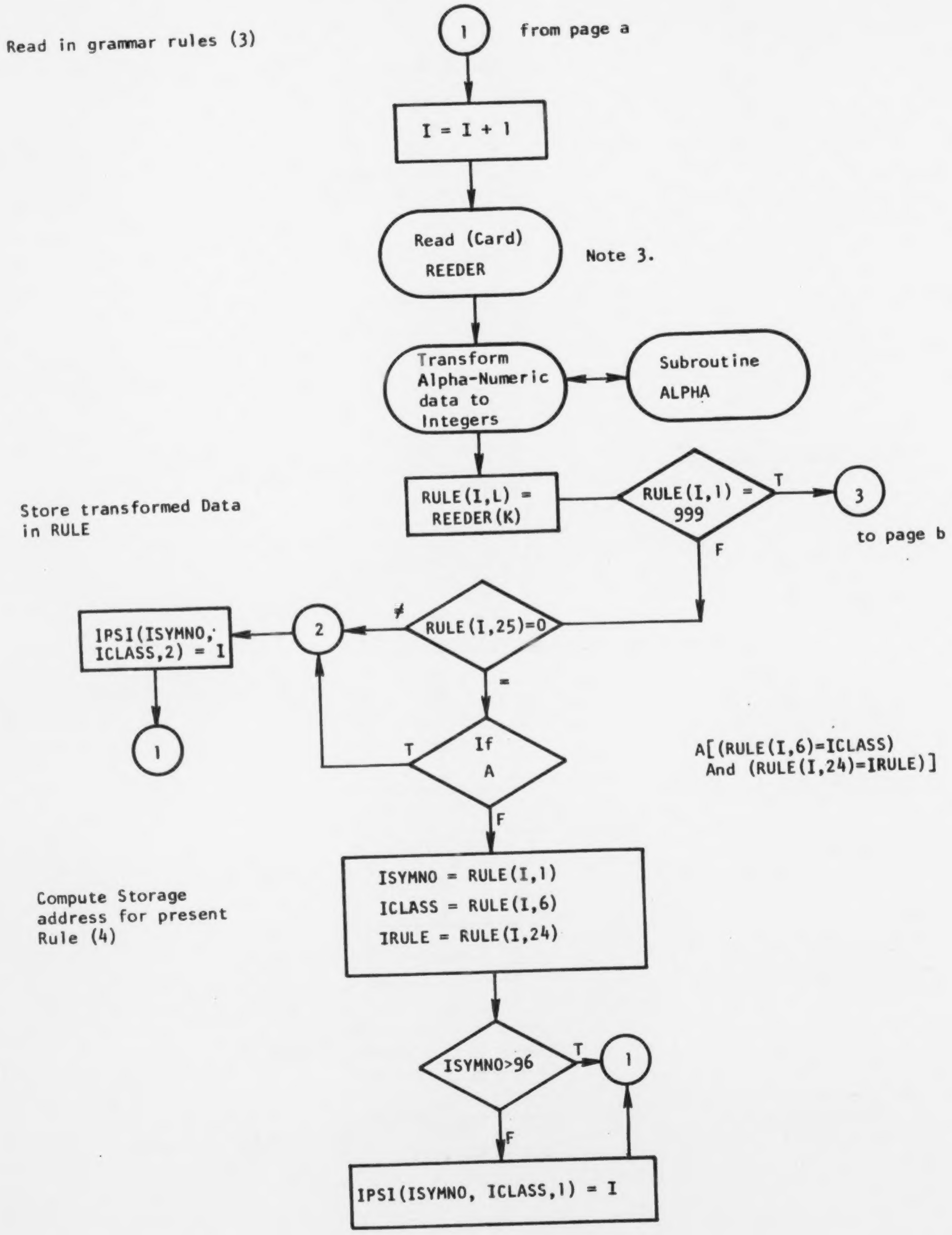

Read in grammar rules (3)
1
from page a
I = I + 1
Read (Card)
REEDER
Note 3.
Transform Alpha-Numeric data to Integers
Subroutine ALPHA
Store transformed Data in RULE
RULE(I,L) = REEDER(K)
RULE(I,1) = 999
T
3
to page b
F
IPSI(ISYMNO, ICLASS,2) = I
2
≠
RULE(I,25)=0
=
1
T
If A
A[(RULE(I,6)=ICLASS) And (RULE(I,24)=IRULE)]
F
Compute Storage address for present Rule (4)
ISYMNO = RULE(I,1)
ICLASS = RULE(I,6)
IRULE = RULE(I,24)
ISYMNO>96
T
1
F
IPSI(ISYMNO, ICLASS,1) = I

Figure 3.3b Main Program SENSYN

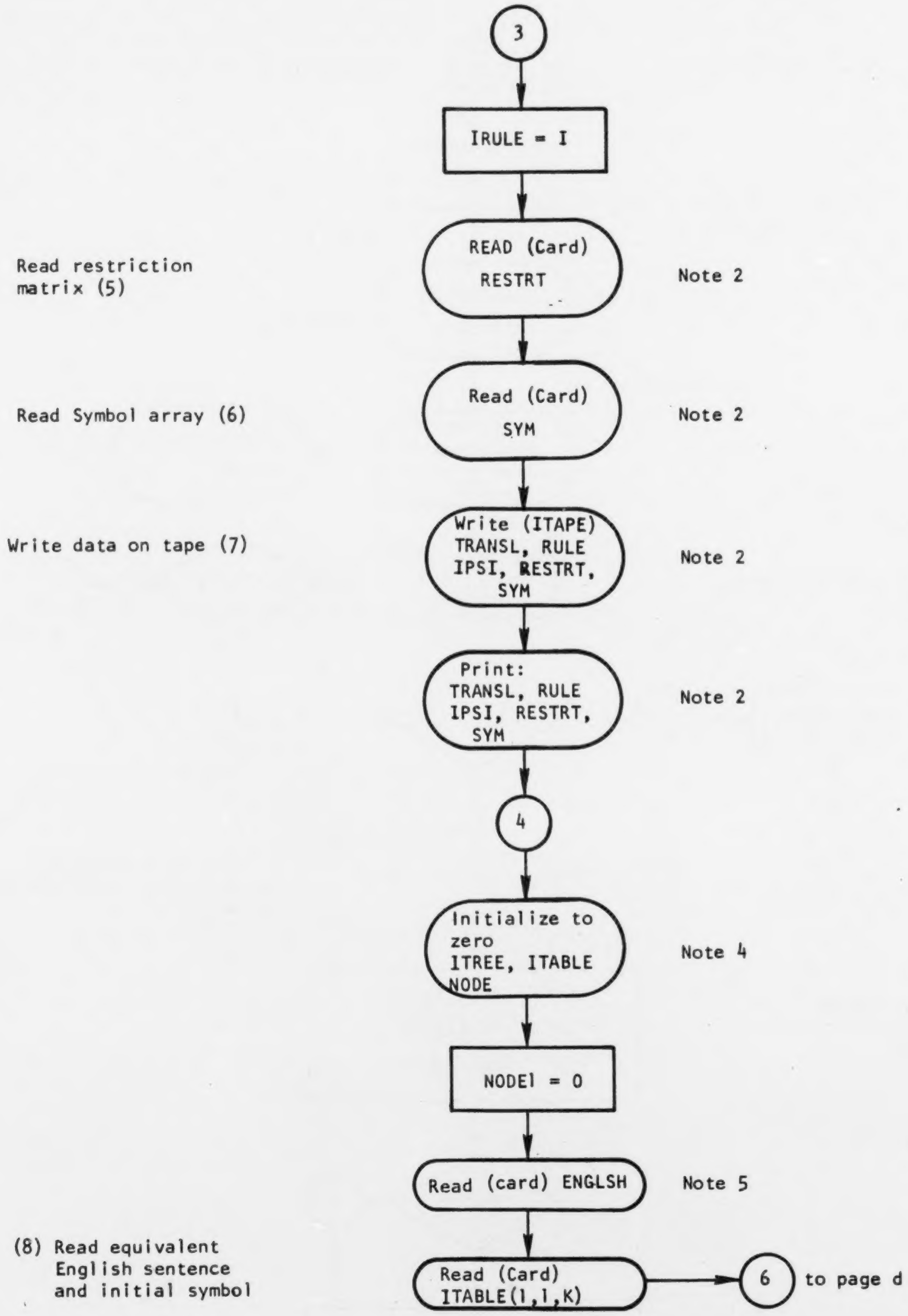

3
IRULE = I
Read restriction matrix (5)
READ (Card) RESTRT
Note 2
Read Symbol array (6)
Read (Card) SYM
Note 2
Write data on tape (7)
Write (ITAPE) TRANSL, RULE IPSI, RESTRT, SYM
Note 2
Print: TRANSL, RULE IPSI, RESTRT, SYM
Note 2
4
Initialize to zero ITREE, ITABLE NODE
Note 4
NODE1 = 0
Read (card) ENGLSH
Note 5
(8) Read equivalent English sentence and initial symbol
Read (Card) ITABLE(1,1,K)
6
to page d

Figure 3.3c Main Program SENSYN

Stop if last sentence (9)

(10) Print out initial symbol

Rewrite J symbols from ITABLE(ILIST,J,K) to ITABLE(ILIST1,L,K) using appropriate rewrite rules (11)

Reset values of ILIST and ILIST1

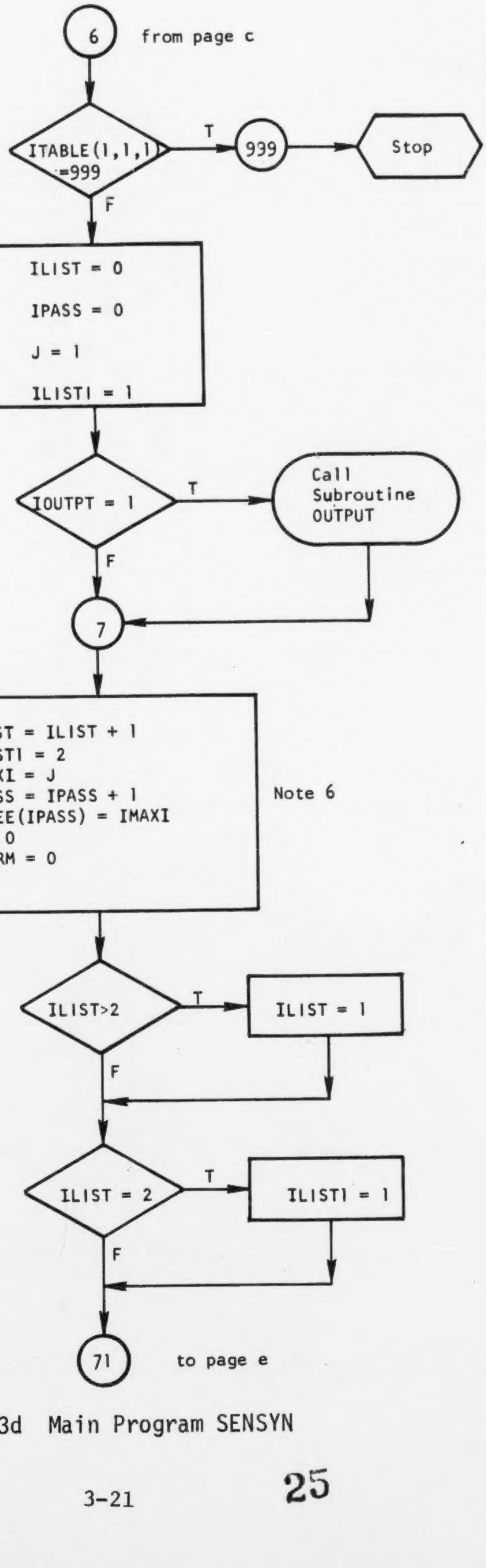

6
from page c
ITABLE(1,1,1) =999
T
999
Stop
F
ILIST = 0
IPASS = 0
J = 1
ILIST1 = 1
IOUTPT = 1
T
Call Subroutine OUTPUT
F
7
ILIST = ILIST + 1
ILIST1 = 2
IMAXI = J
IPASS = IPASS + 1
ITREE(IPASS) = IMAXI
J = 0
ITERM = 0
Note 6
ILIST>2
T
ILIST = 1
F
ILIST = 2
T
ILIST1 = 1
F
71
to page e

Figure 3.3d Main Program SENSYN

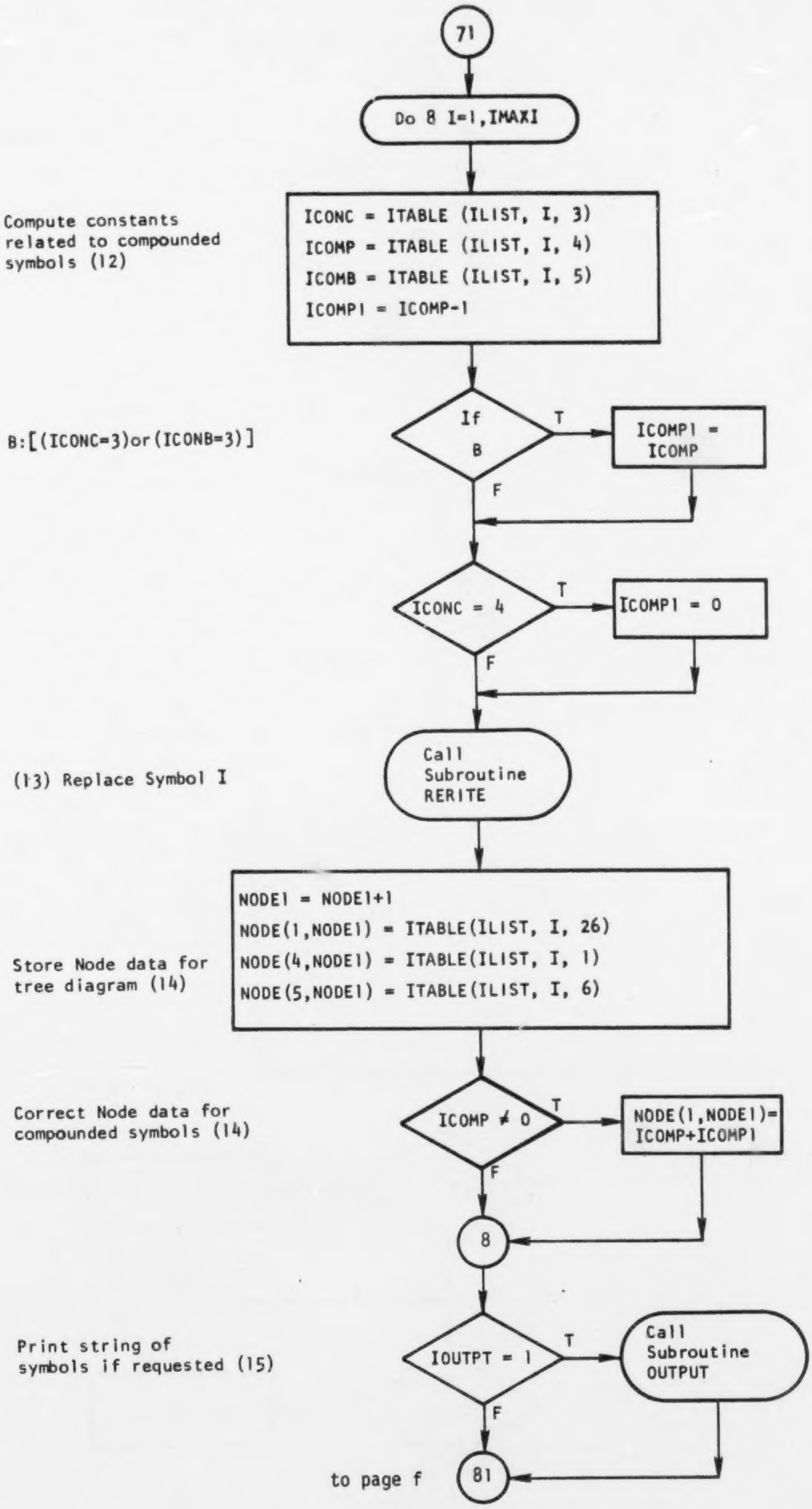

71
Do 8 I=1,IMAXI
Compute constants related to compounded symbols (12)
ICONC = ITABLE (ILIST, I, 3)
ICOMP = ITABLE (ILIST, I, 4)
ICOMB = ITABLE (ILIST, I, 5)
ICOMP1 = ICOMP-1
B:[(ICONC=3)or(ICONB=3)]
If B
T
F
ICOMP1 = ICOMP
ICONC = 4
T
F
ICOMP1 = 0
(13) Replace Symbol I
Call Subroutine RERITE
NODE1 = NODE1+1
NODE(1,NODE1) = ITABLE(ILIST, I, 26)
NODE(4,NODE1) = ITABLE(ILIST, I, 1)
NODE(5,NODE1) = ITABLE(ILIST, I, 6)
Store Node data for tree diagram (14)
Correct Node data for compounded symbols (14)
ICOMP ≠ 0
T
F
NODE(1,NODE1)= ICOMP+ICOMP1
8
Print string of symbols if requested (15)
IOUTPT = 1
T
F
Call Subroutine OUTPUT
to page f
81

Figure 3.3e Main Program SENSYN

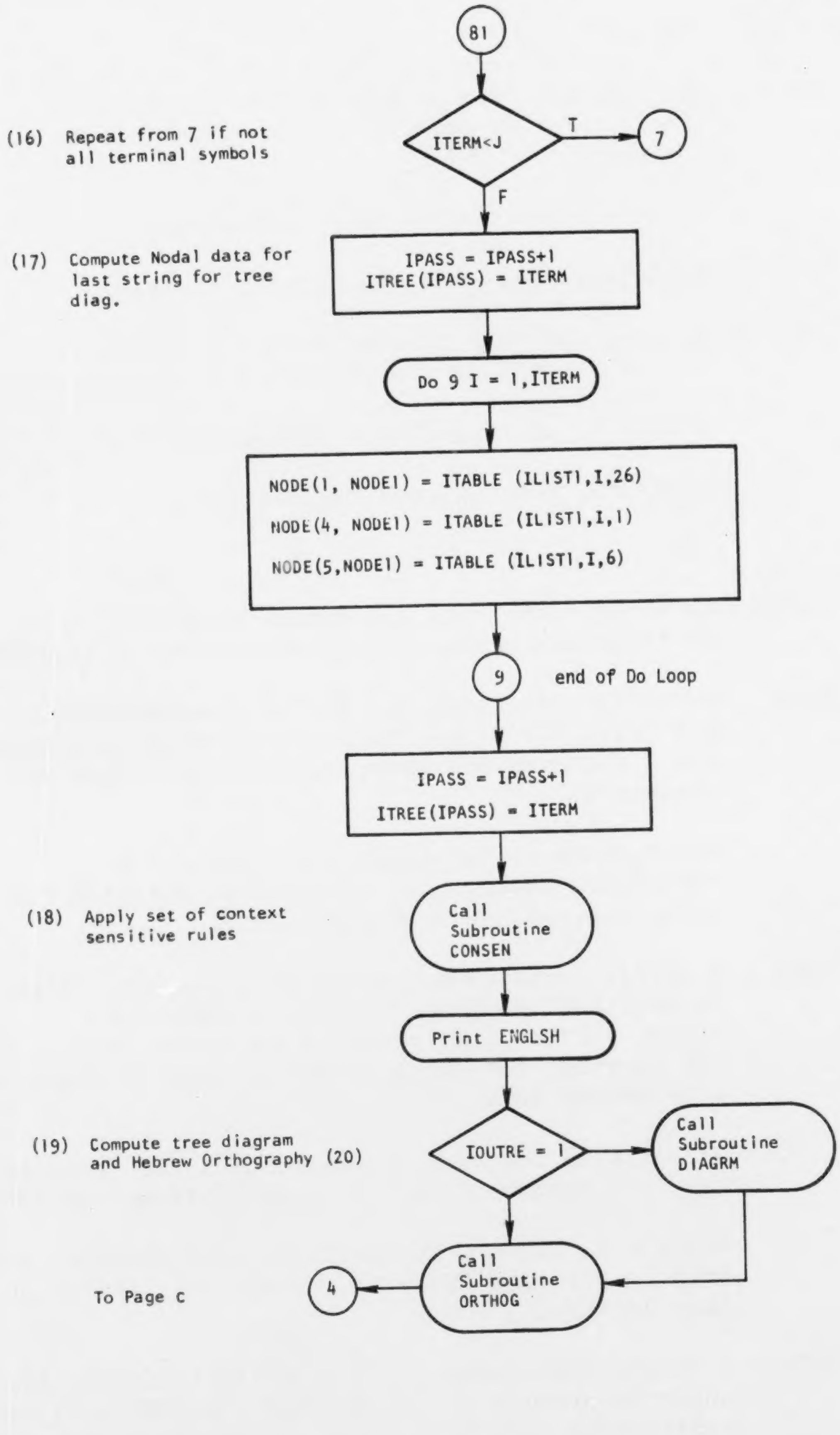

81
(16) Repeat from 7 if not all terminal symbols
ITERM<J
T
7
F
(17) Compute Nodal data for last string for tree diag.
IPASS = IPASS+1
ITREE(IPASS) = ITERM
Do 9 I = 1,ITERM
NODE(1, NODE1) = ITABLE (ILIST1,I,26)
NODE(4, NODE1) = ITABLE (ILIST1,I,1)
NODE(5,NODE1) = ITABLE (ILIST1,I,6)
9
end of Do Loop
IPASS = IPASS+1
ITREE(IPASS) = ITERM
(18) Apply set of context sensitive rules
Call Subroutine CONSEN
Print ENGLSH
(19) Compute tree diagram and Hebrew Orthography (20)
IOUTRE = 1
Call Subroutine DIAGRM
To Page c
4
Call Subroutine ORTHOG

Figure 3.3f Main Program SENSYN

Note 1 (cont.):

IOUTPT:	Enables the user to request the printing of the strings of symbols at every pass of the grammar. IOUTPT=0, no listing of symbol strings IOUTPT≠0, print listing of symbol strings.
IOUTRE:	Enables the user to request the printing of a tree diagram of the generated Hebrew sentences. IOUTRE=0, no tree diagram requested. IOUTRE≠0, tree diagram requested

Note 2:

TRANSL:	An array containing the Hebrew Transliteration, used to transform alpha-numeric characters to integers.
RULE:	A matrix containing the syntax grammar rules. The rules are stored in RULE(i,j) such that each row i contains one symbol with J subscripts in columns j. The symbols of the rules are stored in a continuous string with matrix IPSI defining the beginning and end of a given rule.
IPSI:	A matrix containing the address of syntax rules in matrix RULE based on symbol number and class. IPSI(K,L,1) contains the value of I for locating the first symbol of Rule K, Class L in matrix RULE. IPSI(K,L,2) contains the value of I for locating the last symbol of Rule K, Class L in Matrix RULE. If Rule K, Class L has more than one version, IPSI (K,L,2) contains the value of the last symbol of the last version.
RESTRT:	A matrix containing the list of restrictions that apply to symbols of the grammar. RESTRT(I,J) means restriction type I, Element J. Positive numbers mean that one of J elements must be equal to the value of the specified subscript. Negative numbers mean that the specified subscript must not equal the absolute value of the element.

SYM: An array containing the symbol names associated with a given symbol number.

Note 3: Array REEDER is temporary storage used for computing on rule data before storage in matrix RULE.

Note 4:

ITREE: An array used to store node governing data for constructing tree diagrams.

ITABLE: A matrix used to store two strings of symbols; (1) the present string, and (2) the string being computed.

NODE: A matrix used to compute the symbol connectivity for tree diagrams.

For further description of these variables see subroutines DIAGRM.

Note 5:

ENGLSH: An array containing the sentence in English that is equivalent to the Hebrew sentence being generated. This sentence will be printed in the resultant output.

Note 6:

ILIST: The number of the index of matrix ITABLE containing the string of symbols being operated on by the grammar.

ILIST1: The number of the index of matrix ITABLE on which the results of the operation of the grammar are written (i.e., the new string of resulting symbols.)

I: The index number of the symbol in string ILIST on which the grammar operates.

J: The index of symbols on list ITABLE(ILIST,...)

IMAX: The number of symbols on list ITABLE(ILIST,...)

ITERM: The number of terminal symbols counted on list.

3.3.1.2 Subprogram ALPHA

Subprogram ALPHA (Fig. 3.4) is used to transform alpha-numeric symbols into integers for use in the computations of the program. This subprogram permits input data to be submitted in a form more meaningful to the user. The subprogram functions in the following manner:

(1) A given alpha-numeric symbol (input argument A) is compared with each element in the table of Transliteration (array TRANSL).

(2) When a match is found, an integer value for output argument IA is computed such that IA is one less than the index L.

(3) However, if this value is 40, the value of IA is made zero. This converts all spaces to zero.[7]

The transformation of the data is shown in Table 3-1. Other transformations can be obtained by changing the data in the Table of Transliteration. Subprogram ALPHA is called by the Main Program SENSYN and by Subprogram INPUT.

3.3.1.3 Subprogram ALPHA1

This subprogram (Fig. 3.5) is identical to subprogram ALPHA except that it converts all spaces to the equivalent of *, that is, a value of 40 for IA is changed to 44 in step (3).[8] ALPHA1 is called by Subprogram INFLEX.

3.3.1.4 Subprogram CONSEN[9]

This subprogram (Figure 3.6) is called by the Main Program SENSYN (Step 18) to apply certain context-sensitive rules to the last string of symbols. At this stage in the synthesis of the sentence all

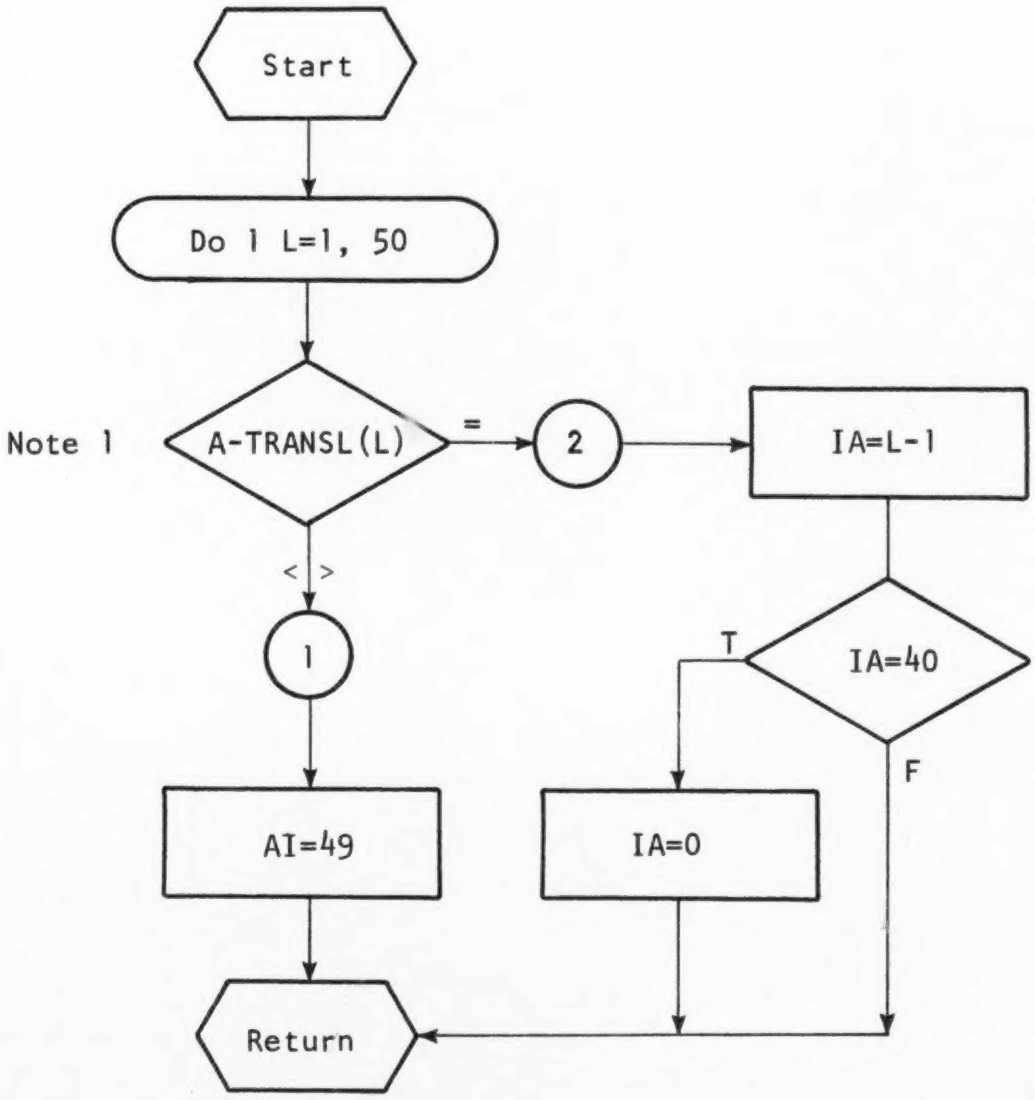
---

[7] This computation is presently dependent on the "content" of the grammar.

[8] This computation is presently dependent on the "content" of the grammar.

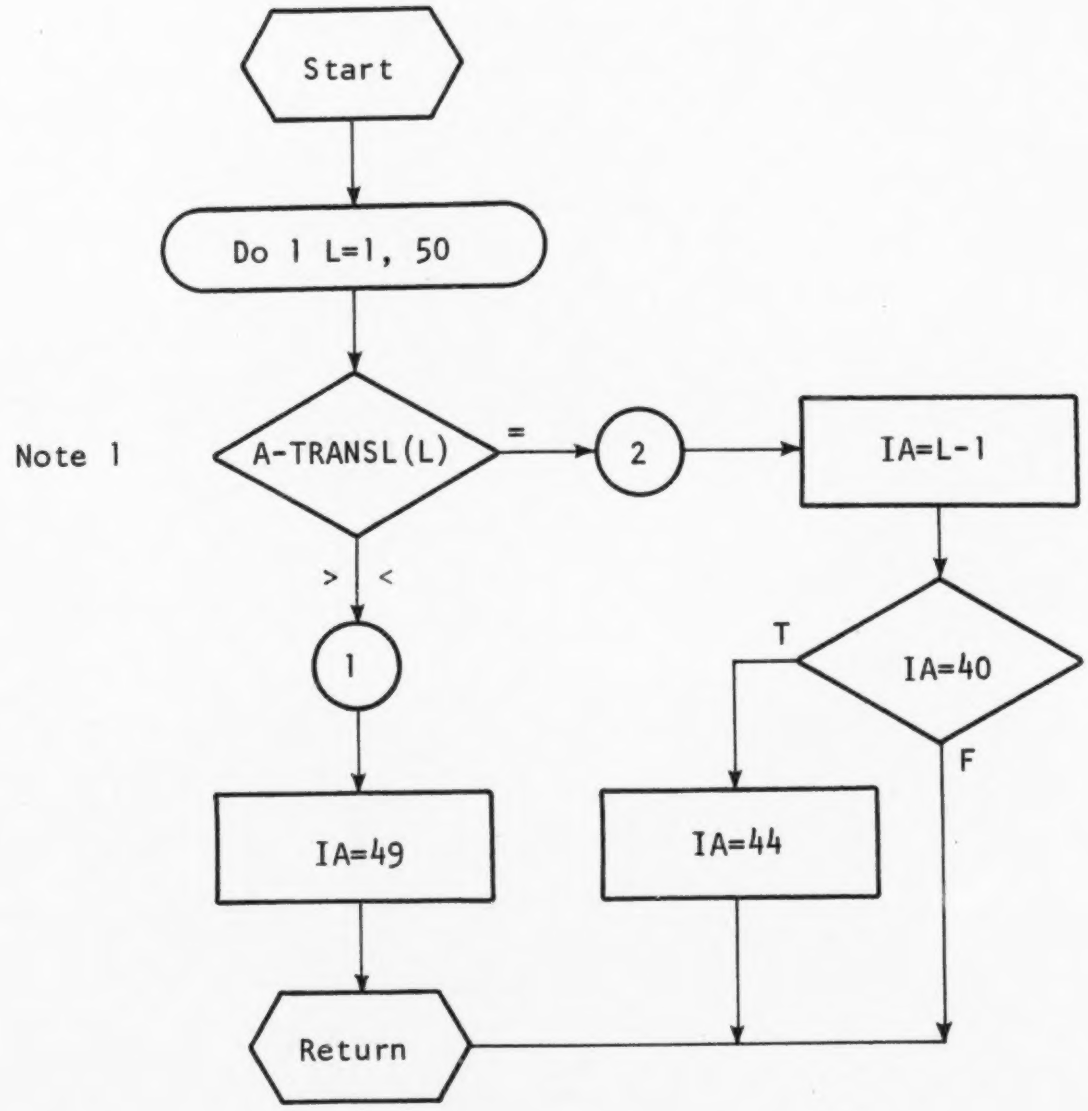
[9] The computations of this subprogram are presently dependent on the "content of the grammar. A future version of the grammar will eliminate the need for this subprogram.

Subroutine ALPHA(A,IA)


Start
Do 1 L=1, 50
Note 1
A-TRANSL(L)
=
2
IA=L-1
< >
1
T
IA=40
F
AI=49
IA=0
Return

Note 1: For TRANSL, see Note 2 of Program SENSYN

Figure 3.4 Subprogram ALPHA


Start
Do 1 L=1, 50
Note 1
A-TRANSL(L)
=
2
IA=L-1
> <
1
T
IA=40
F
IA=49
IA=44
Return

Note 1: For TRANSL, see Note 2 of Program SENSYN

Figure 3.5 Subprogram ALPHA1

symbols in the string are terminal symbols. There are two types of adjustments made to the symbols by the context-sensitive rules:

(1) If the definite article is preceded by the preposition B, K, or L, the definite article is omitted.

(2) If a pronominal suffix appears the preceding word is changed to the construct state with the appropriate value for subscript j.

The subprogram performs the following operations:

(1) Initializes certain indices (Fig. 3.6a)

(2) For each symbol in the string it performs steps 3 through 18 (Fig. 3.6a)

(3) Computes symbol type and class

(4) Skips to step 18 if symbol is not definite article (ITA=83) and not a pronominal suffix (ITA=91, ITC=3) (Fig. 3.6a).

(5) For context sensitive symbols, computes the critical attributes of the preceding symbol (Fig. 3.6a)

(6) If symbol is pronominal suffix branches to step 8 (Fig. 3.6a)

(7) If symbol is definite article, branches to step 17 (Figure 3.6a).

(8) For pronominal suffix, branches to step corresponding to preceding symbol (Fig. 3.6b).

(9) For preceding participle, number, or noun, changes variable h to 1 if fem. pl. (Fig. 3.6b), and skips to step 18.

(10) For pronoun following AYN, makes AYN a construct (j=2) (Figure 3.6b), and skips to step 18.

(11) For pronoun following AWT, makes AWT a construct (j-2)(Fig. 3.6b), and skips to step 18.

(12) For pronoun following preposition, makes preposition a construct with the appropriate value of j corresponding to the preceding preposition (Fig. 3.6c) and skips to step 18.

(13) For pronoun following YS or SL, make YS or SL a construct with appropriate value of *j* (Figure 3.6d), and skips to step 18.

(14) For pronoun following a verb, makes verb a construct (Fig. 3.6d) and skips to step 18.

(15) For pronoun following an infinitive, makes infinitive a construct (Fig. 3.6e), and skips to step 18.

(16) Rewrites the symbol with corrected attributes (Fig. 3.6e), and skips to step 18.

(17) For a definite article following B, K, or L, omits the symbol, and removes symbol from node matrix of tree diagram (Figure 3.6f).

(18) Returns to step 3 for next symbol.

(19) When all symbols are processed returns to calling program.

3.3.1.5 Subprogram DIAGRM

This subprogram (Figure 3.7) is called by the Main Program SENSYN (step 19) to construct a tree diagram of a generated Hebrew sentence. (See Section 2.3.1 of Part II of this report for examples of tree diagrams.) At this stage the synthesis of the sentence is complete and data defining the nodal structure of the tree diagram has been computed and stored in Matrix NODE, and Array ITREE. Matrix NODE contains a row corresponding to each node in the tree diagram. The data in Matrix NODE is as follows:

NODE(I,J):

J = Node number

I = 1, Number of nodes the J-th node governs

I = 2, Line position of J-th node

I = 3, node number of node governing J-th node

I = 4, Symbol type at J-th node

I = 5, Symbol class at J-th node

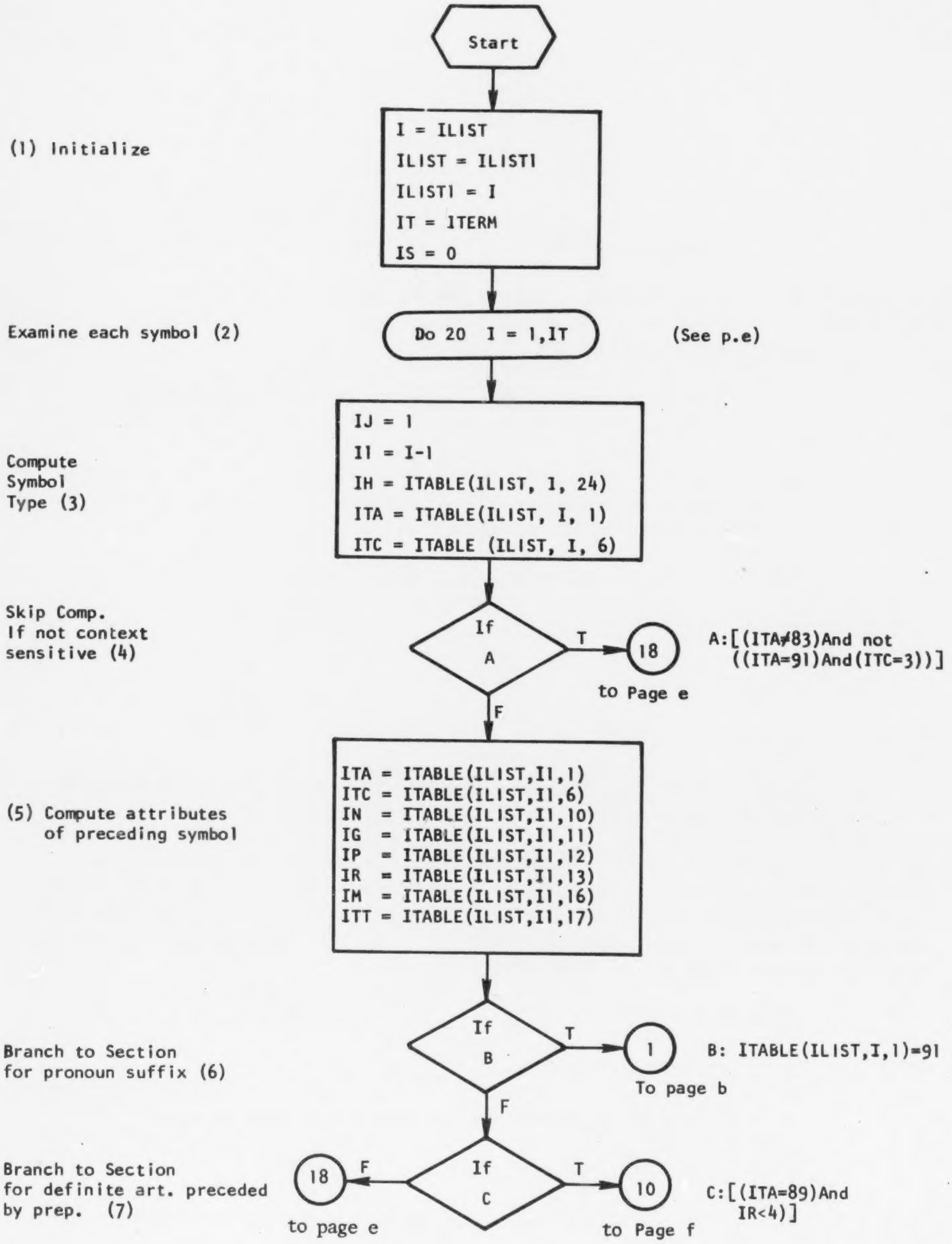

Start
(1) Initialize
I = ILIST
ILIST = ILIST1
ILIST1 = I
IT = ITERM
IS = 0
Examine each symbol (2)
Do 20 I = 1,IT
(See p.e)
Compute Symbol Type (3)
IJ = 1
I1 = I-1
IH = ITABLE(ILIST, I, 24)
ITA = ITABLE(ILIST, I, 1)
ITC = ITABLE (ILIST, I, 6)
Skip Comp. If not context sensitive (4)
If A
T
F
18
to Page e
A:[(ITA≠83)And not ((ITA=91)And(ITC=3))]
(5) Compute attributes of preceding symbol
ITA = ITABLE(ILIST,I1,1)
ITC = ITABLE(ILIST,I1,6)
IN = ITABLE(ILIST,I1,10)
IG = ITABLE(ILIST,I1,11)
IP = ITABLE(ILIST,I1,12)
IR = ITABLE(ILIST,I1,13)
IM = ITABLE(ILIST,I1,16)
ITT = ITABLE(ILIST,I1,17)
Branch to Section for pronoun suffix (6)
If B
T
F
1
To page b
B: ITABLE(ILIST,I,1)=91
Branch to Section for definite art. preceded by prep. (7)
If C
F
T
18
to page e
10
to Page f
C:[(ITA=89)And IR<4)]

Figure 3-6a Subprogram CONSEN

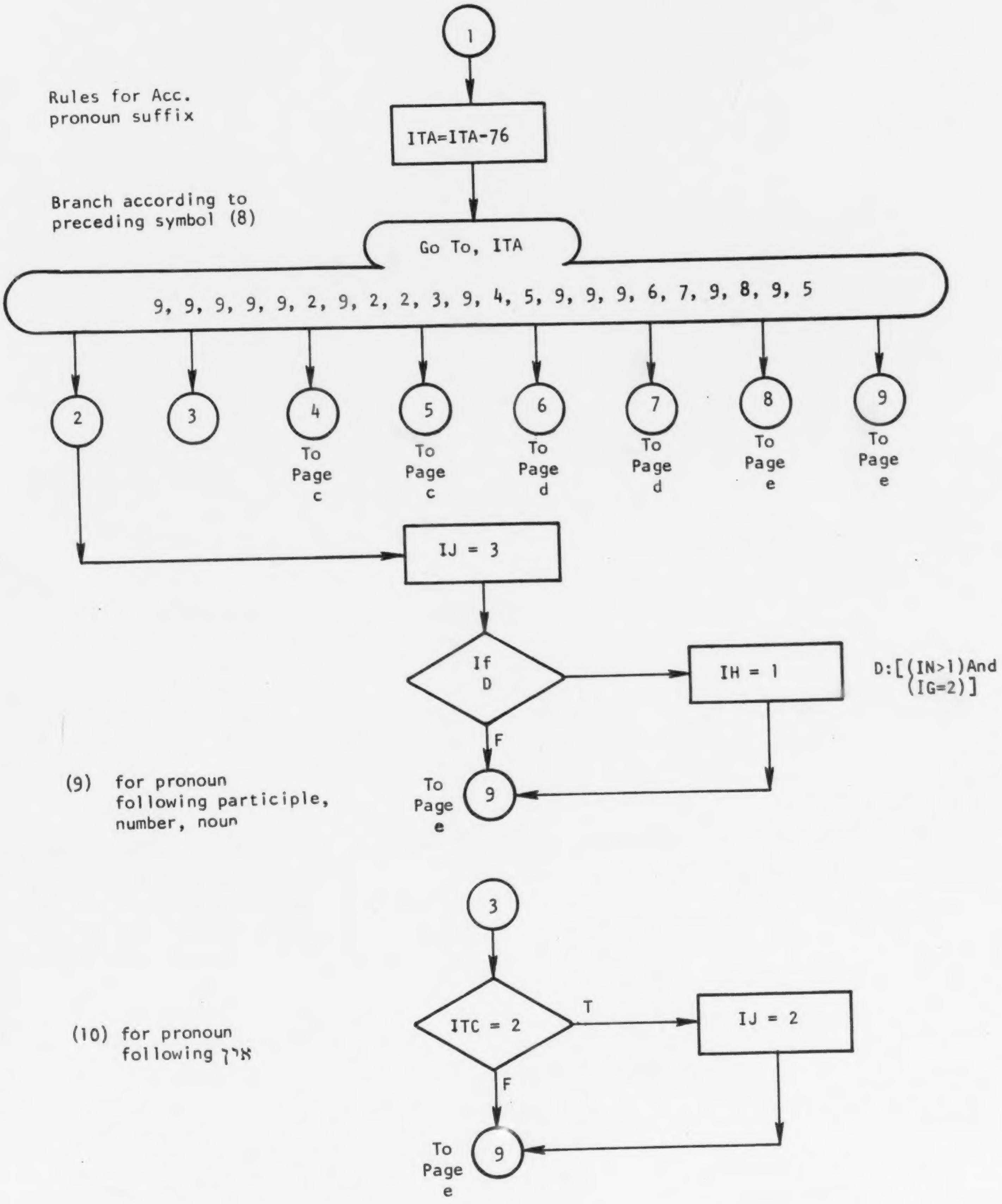
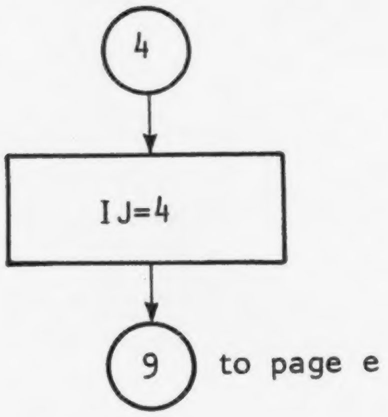
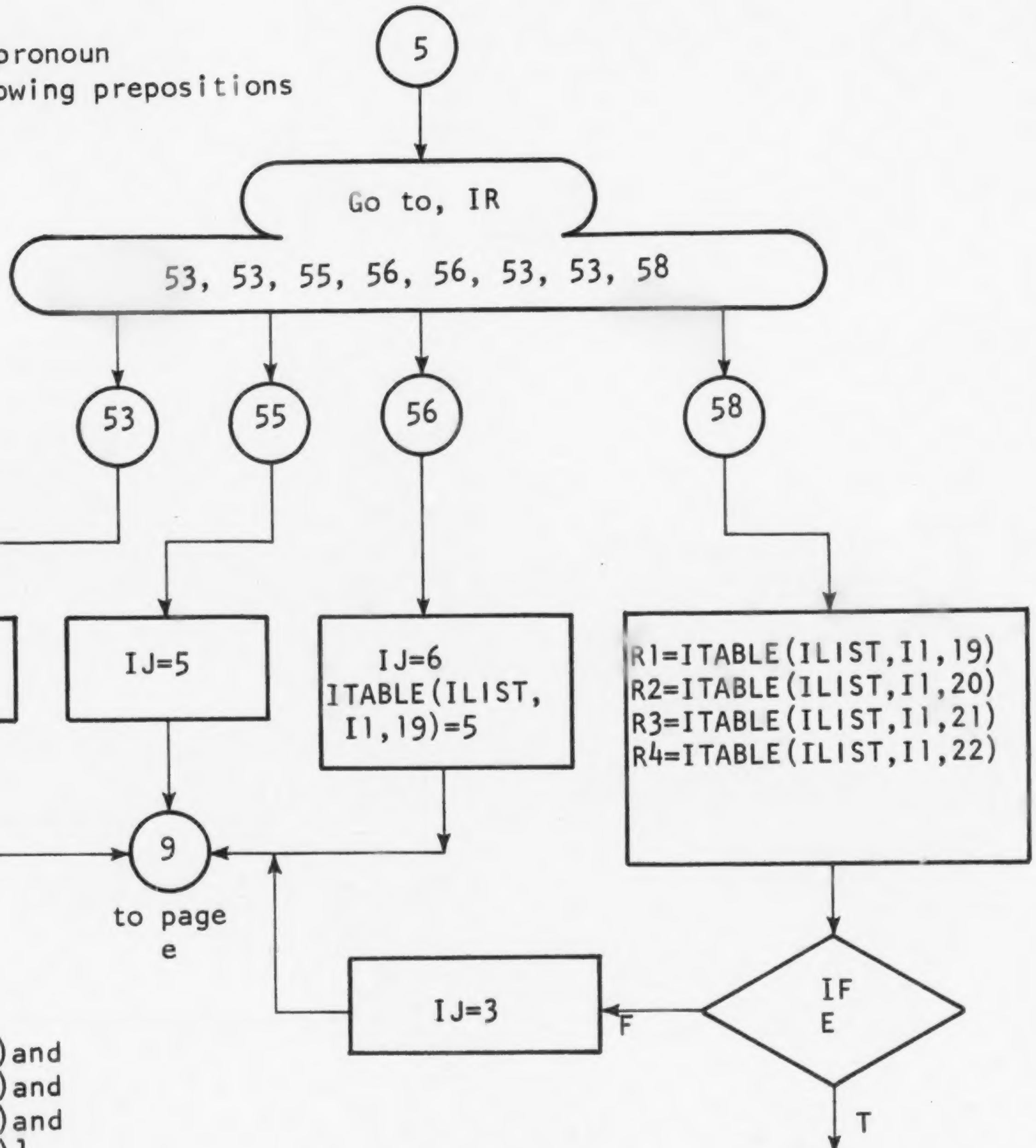

Rules for Acc.
pronoun suffix
1
ITA=ITA-76
Branch according to
preceding symbol (8)
Go To, ITA
9, 9, 9, 9, 9, 2, 9, 2, 2, 3, 9, 4, 5, 9, 9, 9, 6, 7, 9, 8, 9, 5
2
3
4
To Page c
5
To Page c
6
To Page d
7
To Page d
8
To Page e
9
To Page e
IJ = 3
If D
IH = 1
D:[(IN>1)And (IG=2)]
F
(9) for pronoun following participle, number, noun
To Page e
9
3
ITC = 2
T
IJ = 2
(10) for pronoun following אין
F
To Page e
9

Figure 3.6b Subprogram CONSEN

(11) For pronoun following אות

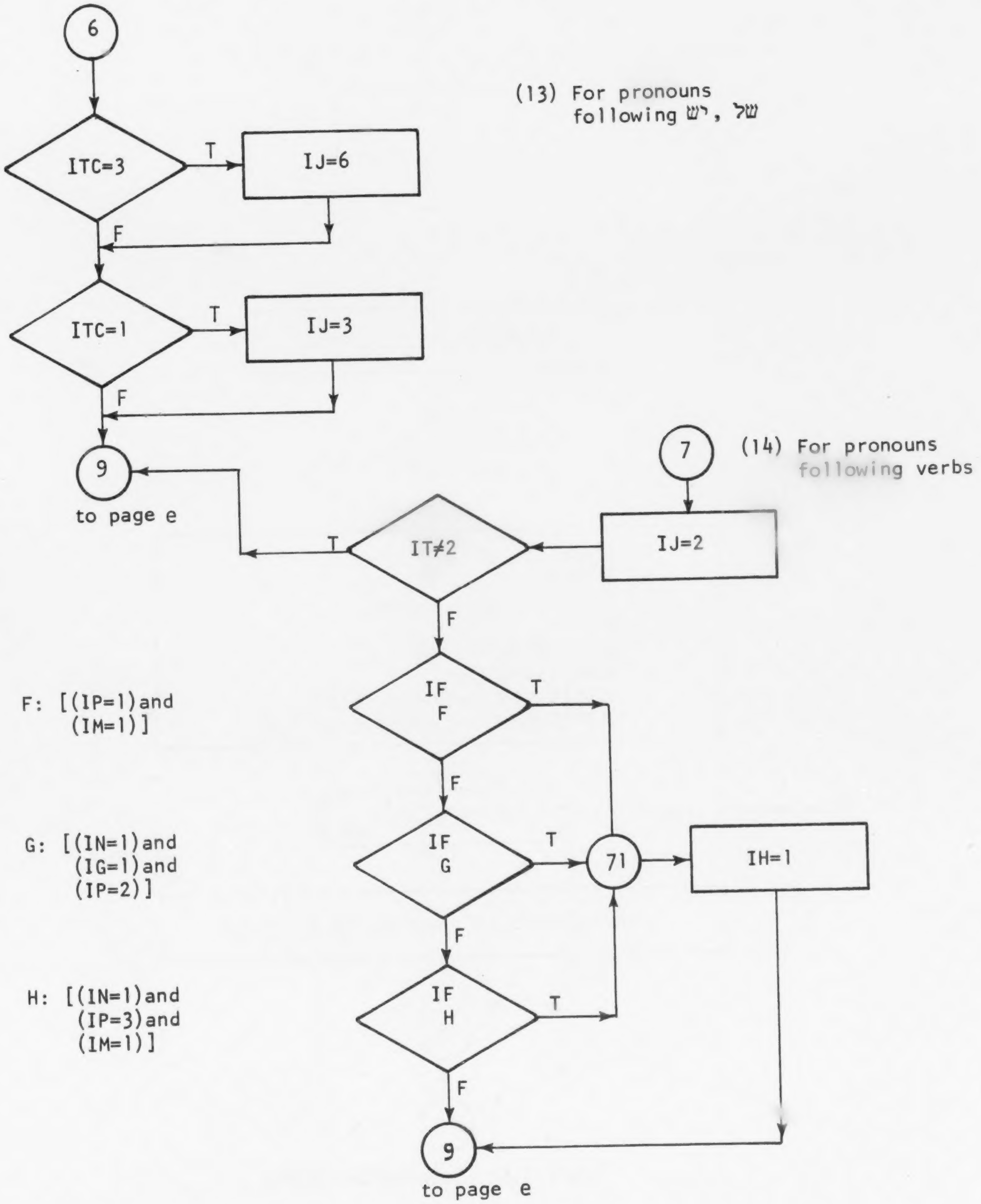

4
IJ=4
9
to page e

(12) For pronoun following prepositions


5
Go to, IR
53, 53, 55, 56, 56, 53, 53, 58
53
55
56
58
IJ=3
IJ=5
IJ=6
ITABLE(ILIST, I1,19)=5
R1=ITABLE(ILIST,I1,19)
R2=ITABLE(ILIST,I1,20)
R3=ITABLE(ILIST,I1,21)
R4=ITABLE(ILIST,I1,22)
9
to page e
IJ=3
F
IF E
T
4

E: [(R1=10)and
(R2=27)and
(R3=21)and
(R4=44)]

Figure 3.6c Subprogram CONSEN



6
ITC=3
T
IJ=6
F
ITC=1
T
IJ=3
F
9
to page e
7
IJ=2
IT≠2
T
F
IF F
T
F
IF G
T
71
IH=1
F
IF H
T
F
9
to page e

(13) For pronouns following יש, של

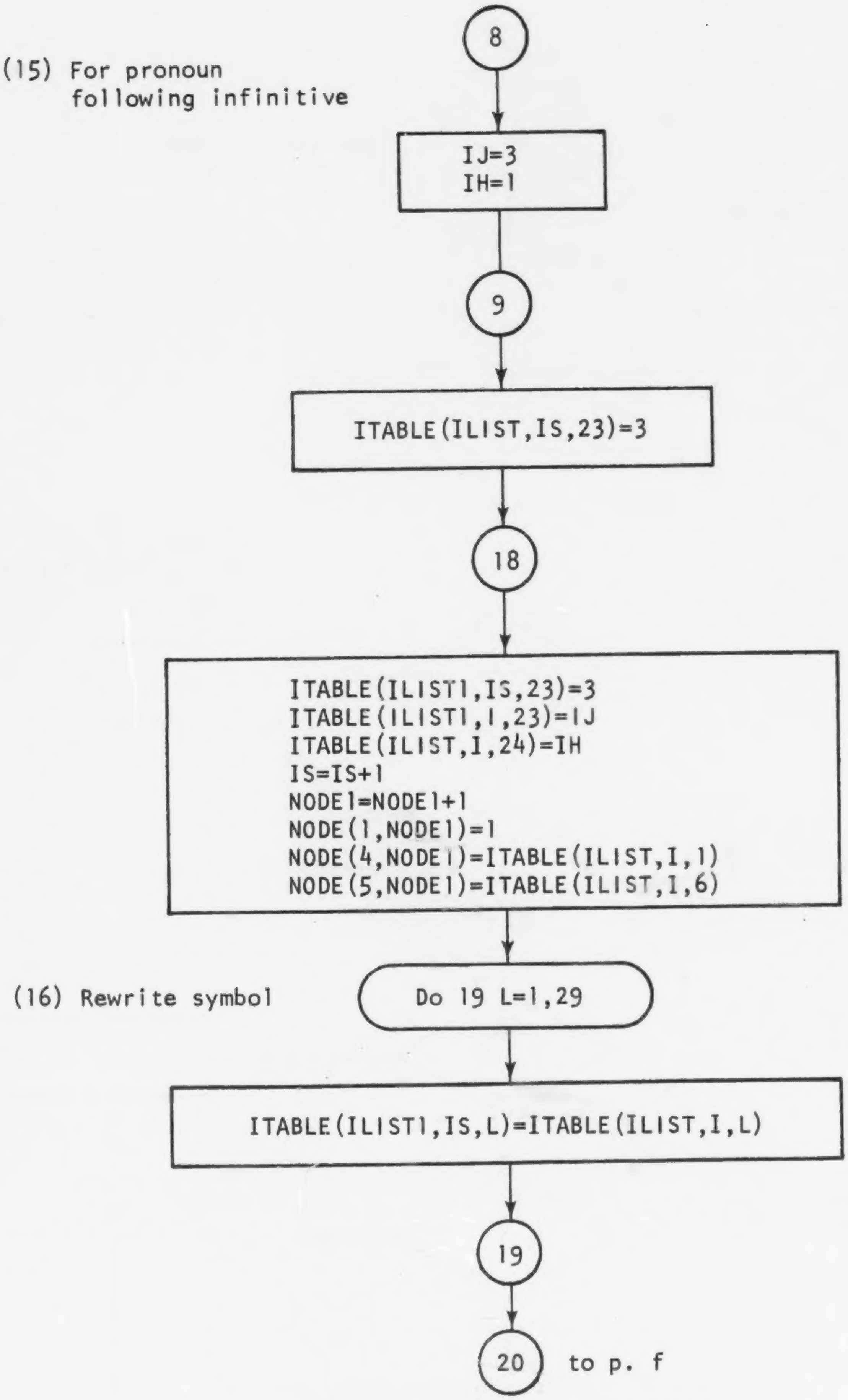
(14) For pronouns following verbs

F: [(IP=1)and (IM=1)]

G: [(IN=1)and (IG=1)and (IP=2)]

H: [(IN=1)and (IP=3)and (IM=1)]

Figure 3.6d Subprogram CONSEN


8
(15) For pronoun following infinitive
9
18
(16) Rewrite symbol
19
20
to p. f

```
IJ=3
IH=1
```

```
ITABLE(ILIST,IS,23)=3
```

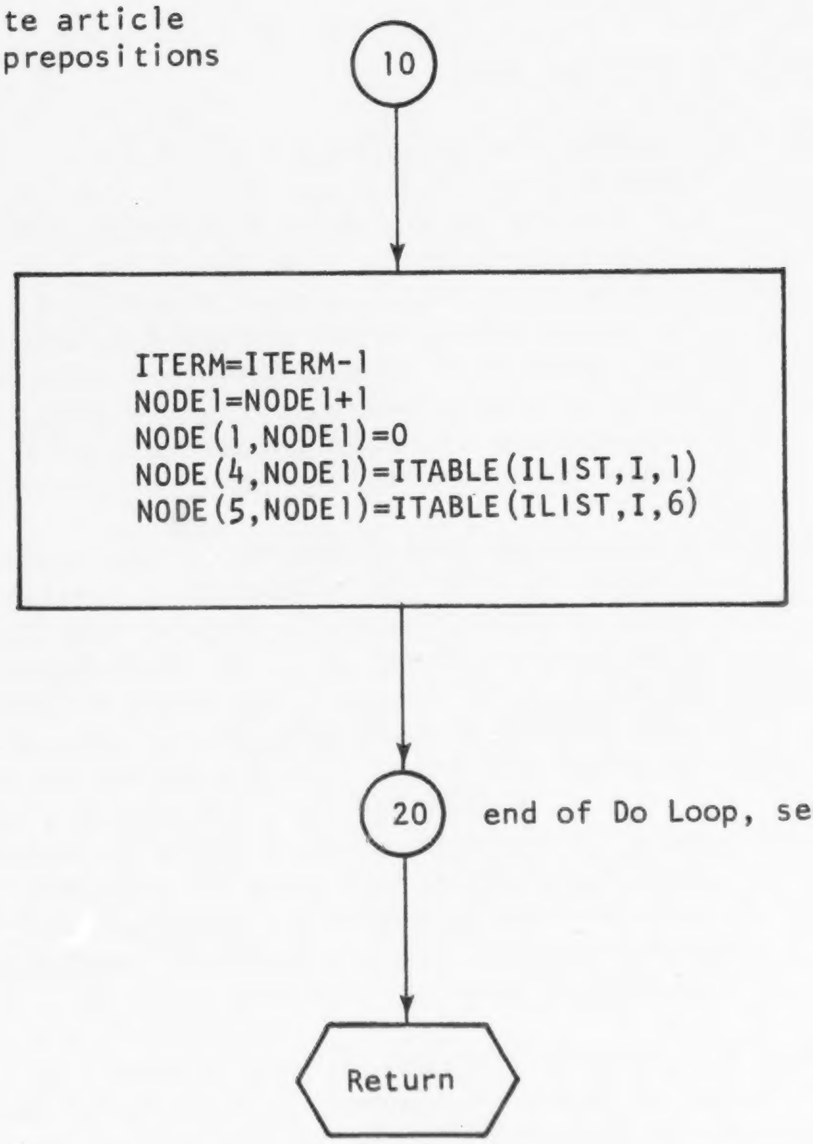
```
ITABLE(ILIST1,IS,23)=3
ITABLE(ILIST1,I,23)=IJ
ITABLE(ILIST,I,24)=IH
IS=IS+1
NODE1=NODE1+1
NODE(1,NODE1)=1
NODE(4,NODE1)=ITABLE(ILIST,I,1)
NODE(5,NODE1)=ITABLE(ILIST,I,6)
```

```
Do 19 L=1,29
```

```
ITABLE(ILIST1,IS,L)=ITABLE(ILIST,I,L)
```

Figure 3.6e Subprogram CONSEN

(17) For definite article following prepositions
ל, כ, ב



```
ITERM=ITERM-1
NODE1=NODE1+1
NODE(1,NODE1)=0
NODE(4,NODE1)=ITABLE(ILIST,I,1)
NODE(5,NODE1)=ITABLE(ILIST,I,6)
```

(18) end of Do Loop, see p. a

(19)

Figure 3.6f Subprogram CONSEN

Array ITREE defines the number of nodes at each level of the tree as follows:

ITREE (I):

I = level number

ITREE(I)= number of nodes in I-th level.

Figure 3.7a outlines the five main operations of this subprogram and refers to the corresponding figure for the detailed flow diagram of that operation.

These operations are:

(1) Compute the governing node of each node in the matrix (Figure 3.7b). This operation consists of filling in the data for Matrix NODE(3,J). Computation starts with the first node. Using the data in NODE(1,1) which specifies the number of nodes (N) governed by the first node, the value of 1 is placed in the next N nodes. If a node is found that governs zero nodes, adjustments are made to the data of governing nodes to remove this branch of the tree.

(2) Compute the line position of each node. (Figure 3.7c). This operation consists of filling in the data for Matrix NODE(2,J). Computation starts with the last level of nodes each of which is assigned a line position beginning with position 1 for the first, position 2 for the second, and so forth up to a maximum of 20. Computation then goes to the next to last level of nodes and the position of each node is computed to be midway between the position of the nodes it governs. Successively higher levels are computed until the first node is reached.

Steps 3 through 5 (following) are repeated in sequence for each level of nodes in the tree diagram. These steps cause the computer to print out the tree diagram of the generated sentence one level at a time beginning with the first level. Three parts are required for each level: (a) the upper connectors that show the relationship of the nodes in a given level to the governing nodes at the next highest level; (b) the symbol name and class of each node; and (c) the lower connectors that show the governing relationship of a given node to the nodes at the next lowest level.

(3) Write upper connectors (Figure 3.7d through 3.7f). The upper connectors occupy two lines on the printed output. The first one connects together (with a horizontal line) nodes governed by a common higher level node. The second line provides a vertical bar above each nodes position. This is accomplished by using various combinations of the following 3-character words:

BLANK = ΔΔΔ

DASHES = ---

BAR = IΔΔ

Where Δ represents a space. The logic for the first line is as follows:

(a) If the governing node governs only one node, BLANK + BAR is written above the node position, that is ΔΔΔIΔΔ.

(b) If the governing node governs more than one node, BLANK + DASHES (ΔΔΔ---) is written above the first of the governed nodes, DASHES + DASHES (------) is written above intermediate governed nodes, and DASHES + BLANK (---ΔΔΔ) is written above the last of the governed nodes. This provides a continuous line of dashes from the center of the position of the first node to the center of the position of the last node that are governed by one common higher level node. For the second line of the upper connectors, BLANK + BAR (ΔΔΔIΔΔ) is written above each node that governs at least 1 node and BLANK + BLANK (ΔΔΔΔΔΔ) is written above each node that governs zero nodes.

Upper connectors are omitted for the first level of nodes.

(4) Write line of nodes (Figure 3.7g). This operation consists of printing the symbol name and class at each nodal position in the line. The symbol type is specified by NS=NODE (4,J), and the corresponding symbol name is obtained from SYM(NS). The symbol class is

specified by NT=NODE(5,J) which is transformed to alpha-numeric characters by CLS(NT). However, if the symbol and class of a given node are the same as the symbol and class of the governing node, BLANK + BAR (ΔΔΔIΔΔ) is written in the position of the node; this eliminates redundant data from the tree.

(5) Write lower connectors (Figure 3.7h). This operation consists of writing BLANK + BAR (ΔΔΔIΔΔ) under the position of a given node if it governs one or more nodes and BLANK & BLANK (ΔΔΔΔΔΔ) if it governs zero nodes. This is the same as line 2 of the upper connectors, so the same coding is used for both under control of an index (IUPLOW). The lower connectors are omitted for the last level of nodes.

3.3.1.6 Subprogram INPUT

This subprogram (Figure 3.8) is called by Subprogram RULRIT to read in new input data as required by the rules of the grammar. In the process, the subprogram checks the validity of the data.

The following operations are performed in INPUT:

(1) Initialize the input data register INREAD and the logic monitor MATCH.

(2) Skip step 3 if the "present constituent" does not match the "present rule symbol", and if the "present rule symbol" is not a variable symbol.

(3) Check the data in the "present constituent" register SYMIN for independent variables (SYMIN(L)=9) and transfers these variables to the "present rule symbol" register SYMBOL.

(4) Read an input data card if the "present rule symbol" has an independent variable (SYMBOL(L)=9), otherwise skip to step 7.

(5) Transform the input data to integer form using subprogram ALPHA.

(6) If the symbol type of the input data does not match the "present rule symbol", and if

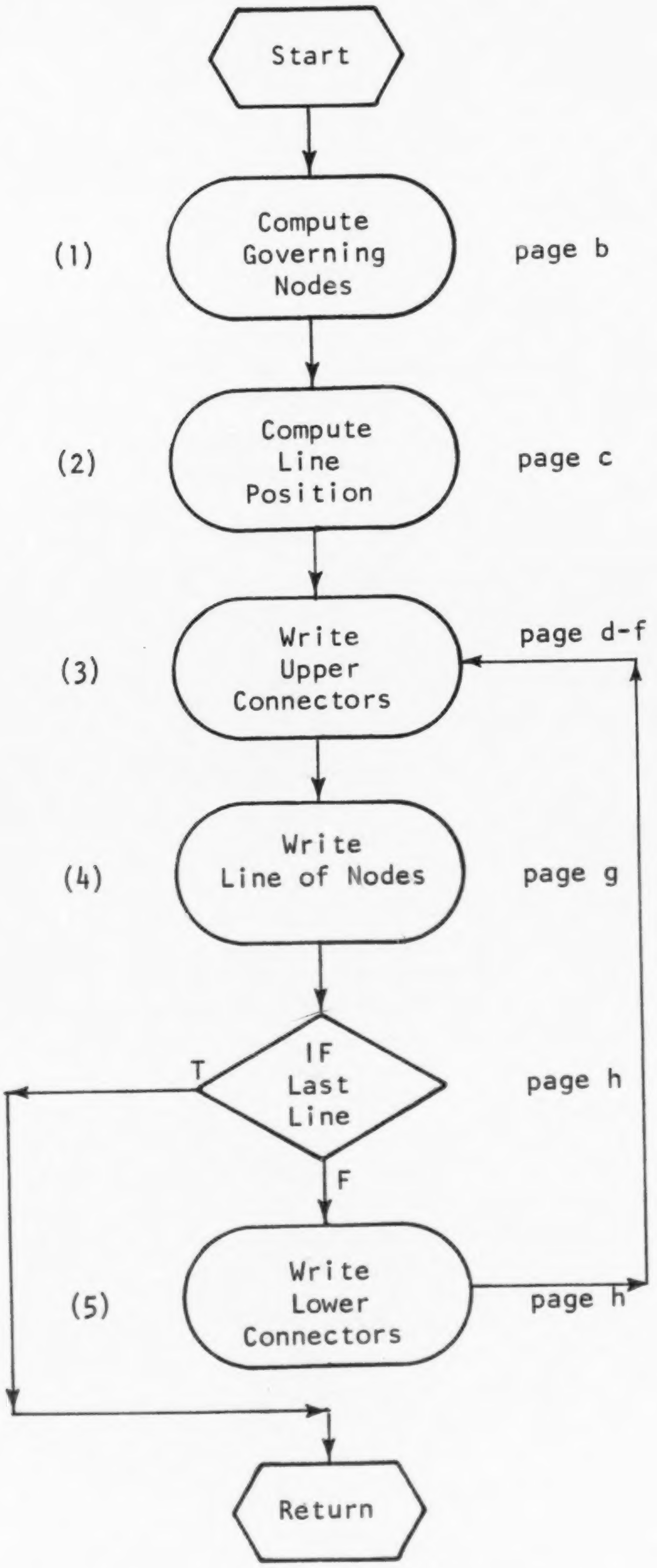

Start
(1)
Compute Governing Nodes
page b
(2)
Compute Line Position
page c
(3)
Write Upper Connectors
page d-f
(4)
Write Line of Nodes
page g
IF Last Line
T
F
page h
(5)
Write Lower Connectors
page h
Return

Figure 3.7a Subprogram DIAGRM

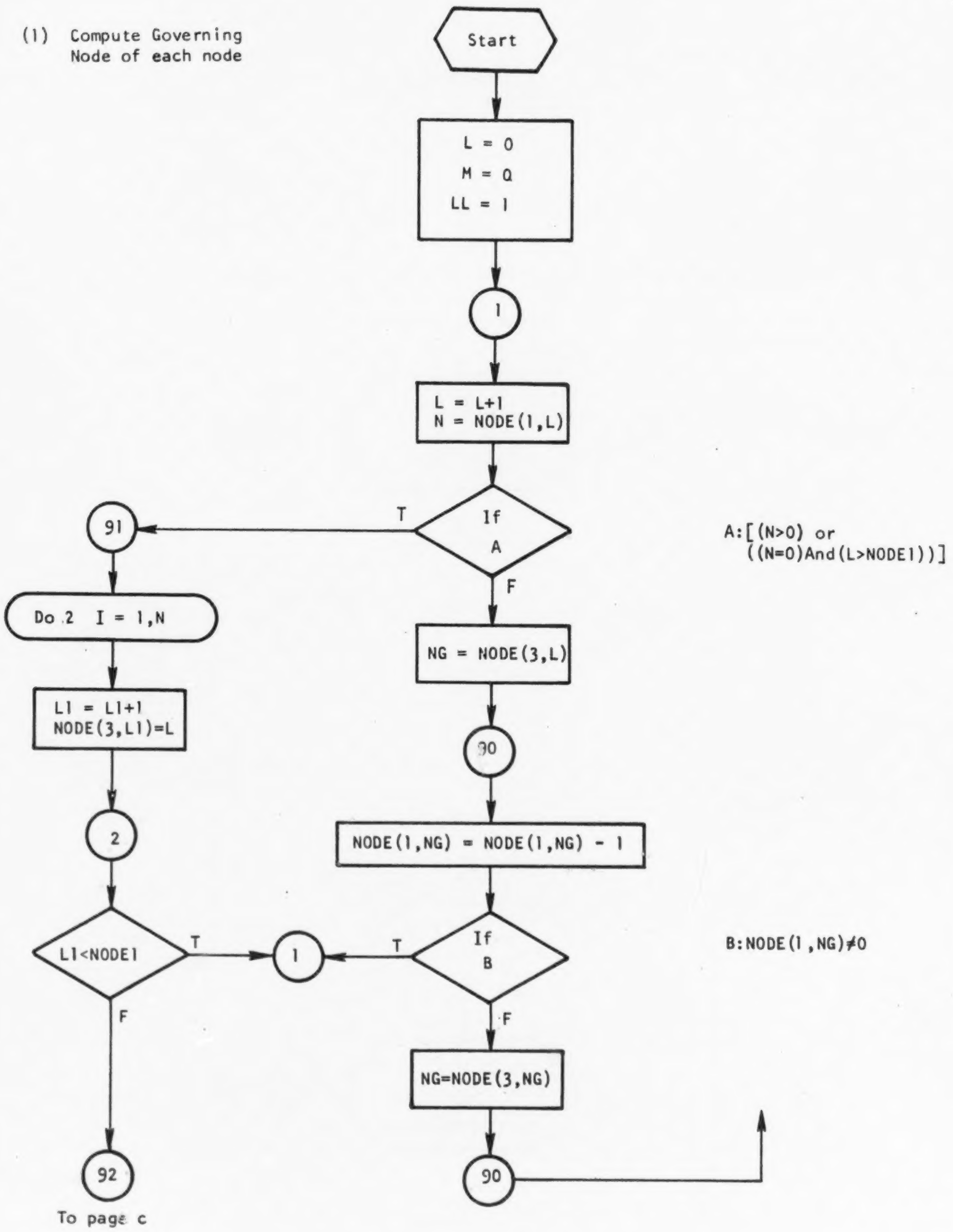
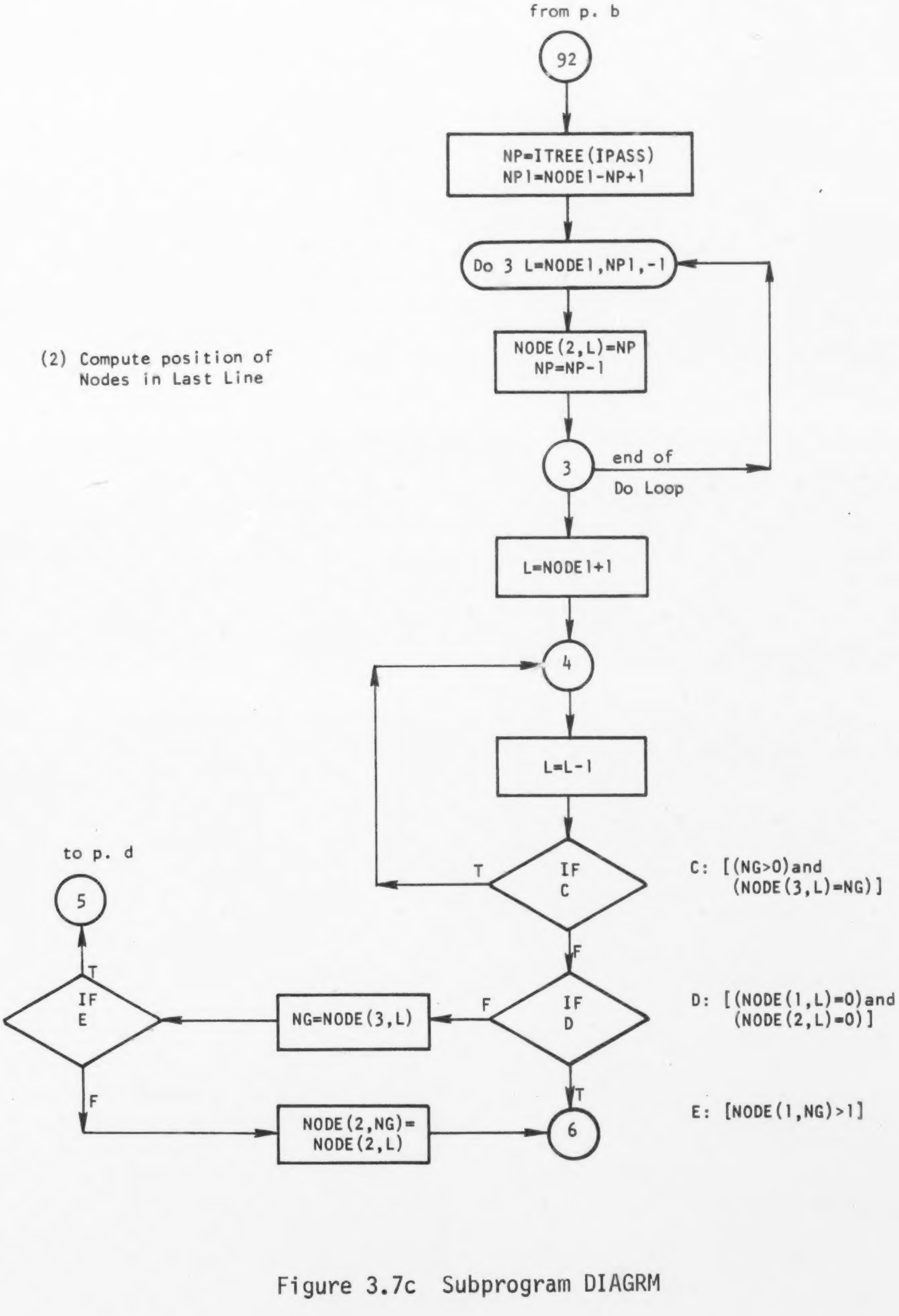

(1) Compute Governing Node of each node
Start
L = 0
M = 0
LL = 1
1
L = L+1
N = NODE(1,L)
If A
T
F
A:[(N>0) or ((N=0)And(L>NODE1))]
91
Do 2 I = 1,N
L1 = L1+1
NODE(3,L1)=L
2
L1<NODE1
T
F
92
To page c
NG = NODE(3,L)
90
NODE(1,NG) = NODE(1,NG) - 1
If B
T
F
1
B:NODE(1,NG)≠0
NG=NODE(3,NG)
90

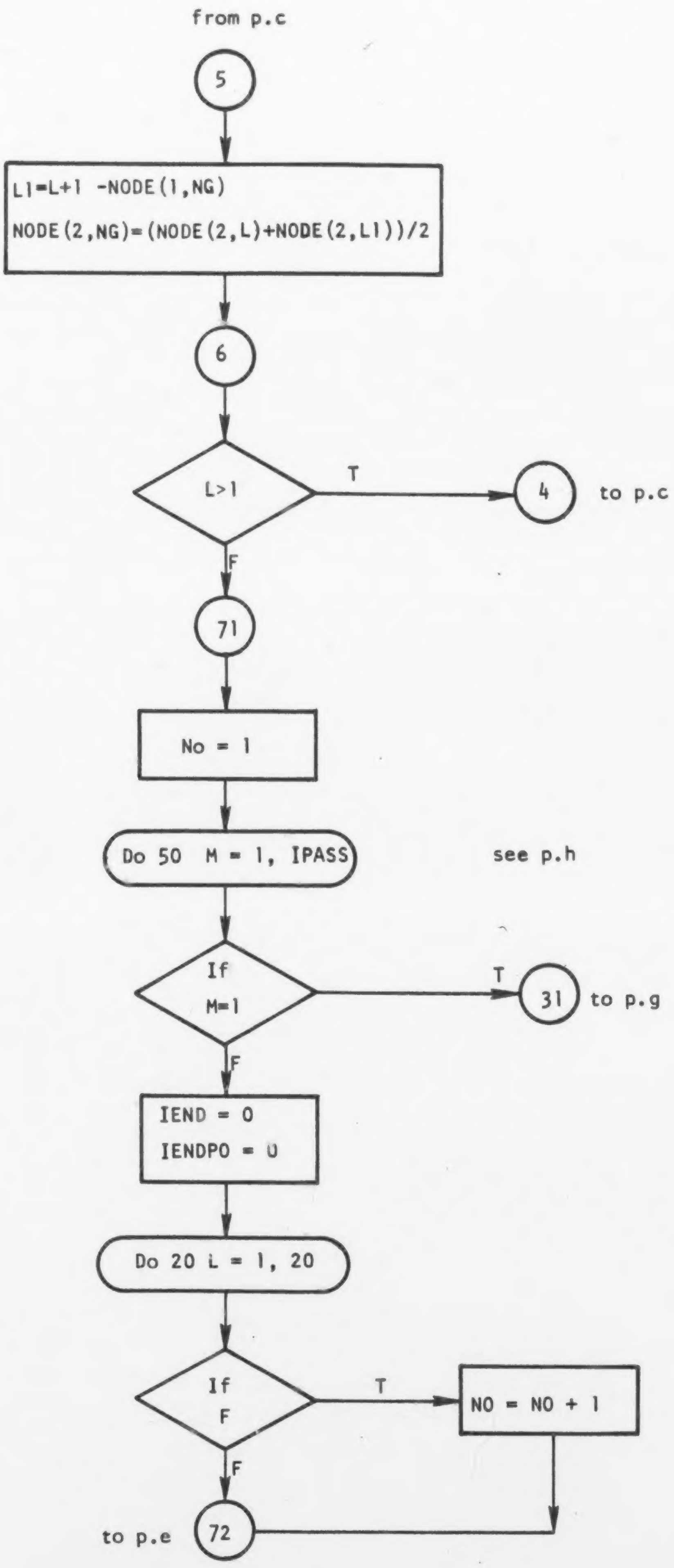
Figure 3.7b Subprogram DIAGRM

(2) Compute position of
Nodes in Last Line



from p. b
92
NP=ITREE(IPASS)
NP1=NODE1-NP+1
Do 3 L=NODE1,NP1,-1
NODE(2,L)=NP
NP=NP-1
3
end of
Do Loop
L=NODE1+1
4
L=L-1
IF C
T
F
C: [(NG>0)and (NODE(3,L)=NG)]
to p. d
5
IF D
D: [(NODE(1,L)=0)and (NODE(2,L)=0)]
NG=NODE(3,L)
IF E
T
F
NODE(2,NG)= NODE(2,L)
6
E: [NODE(1,NG)>1]

Figure 3.7c Subprogram DIAGRM


from p.c
5
L1=L+1 -NODE(1,NG)
NODE(2,NG)=(NODE(2,L)+NODE(2,L1))/2
6
L>1
T
4
to p.c
F
71
No = 1
Do 50 M = 1, IPASS
see p.h
If
M=1
T
31
to p.g
F
IEND = 0
IENDPO = 0
Do 20 L = 1, 20
If
F
T
NO = NO + 1
F
to p.e
72

Governing node put mid-way between first and last governed node

Continue if last node, otherwise repeat for next.

Produce IPASS lines of nodes

Omit upper connectors for first line

(3) Write upper connectors (horizontal)

F:[[(NODE(1,NO)=0) Or (NODE(2,NO)=0)]And [NO≤NODE1]]

Figure 3.7d Subprogram DIAGRM

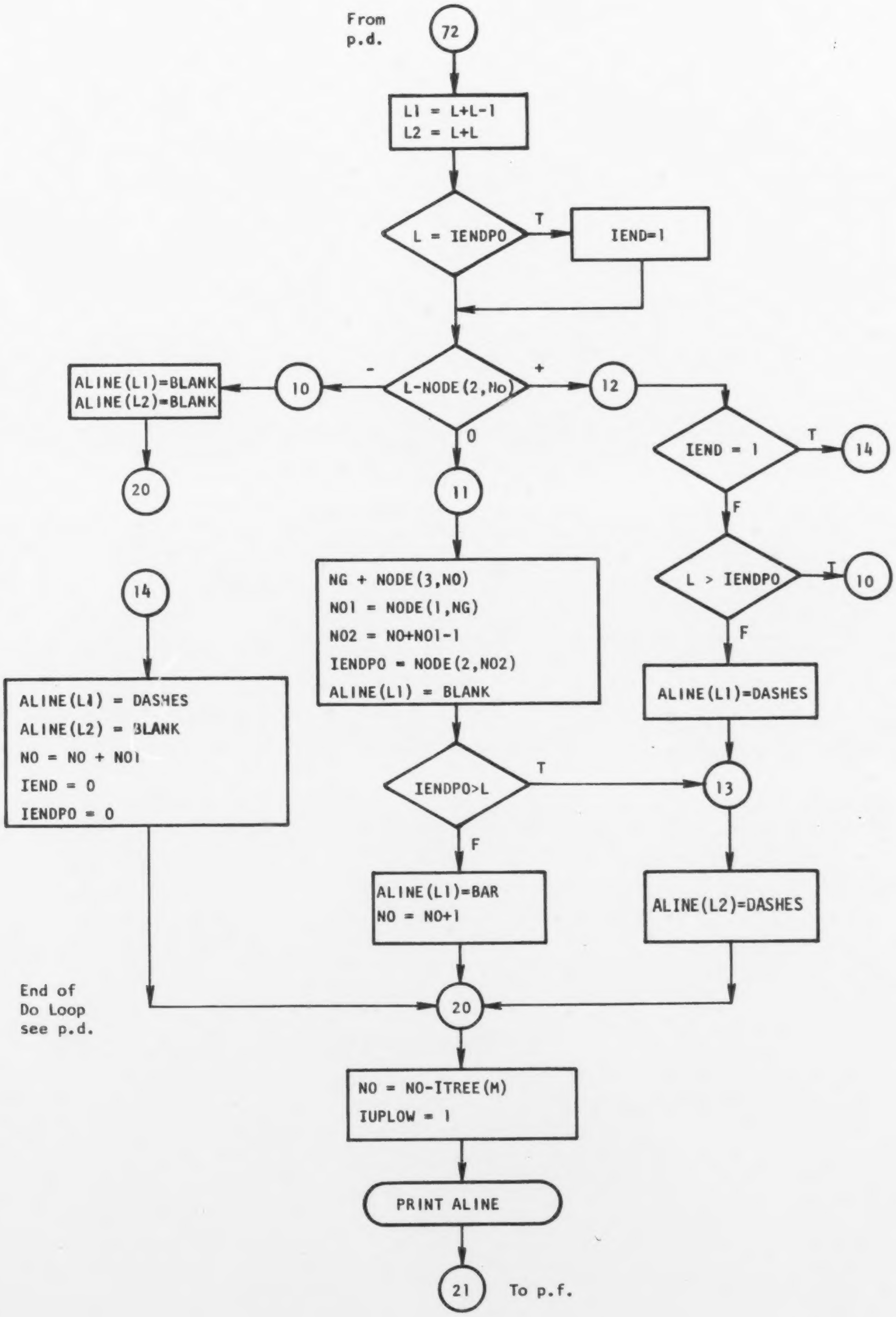

From p.d.
72
L1 = L+L-1
L2 = L+L
L = IENDPO
T
IEND=1
L-NODE(2,No)
-
+
0
10
12
ALINE(L1)=BLANK
ALINE(L2)=BLANK
20
11
IEND = 1
T
14
F
L > IENDPO
T
10
F
14
NG + NODE(3,NO)
NO1 = NODE(1,NG)
NO2 = NO+NO1-1
IENDPO = NODE(2,NO2)
ALINE(L1) = BLANK
ALINE(L1)=DASHES
ALINE(L1) = DASHES
ALINE(L2) = BLANK
NO = NO + NO1
IEND = 0
IENDPO = 0
IENDPO>L
T
13
F
ALINE(L1)=BAR
NO = NO+1
ALINE(L2)=DASHES
End of Do Loop see p.d.
20
NO = NO-ITREE(M)
IUPLOW = 1
PRINT ALINE
21
To p.f.

Figure 3.7e Subprogram DIAGRM

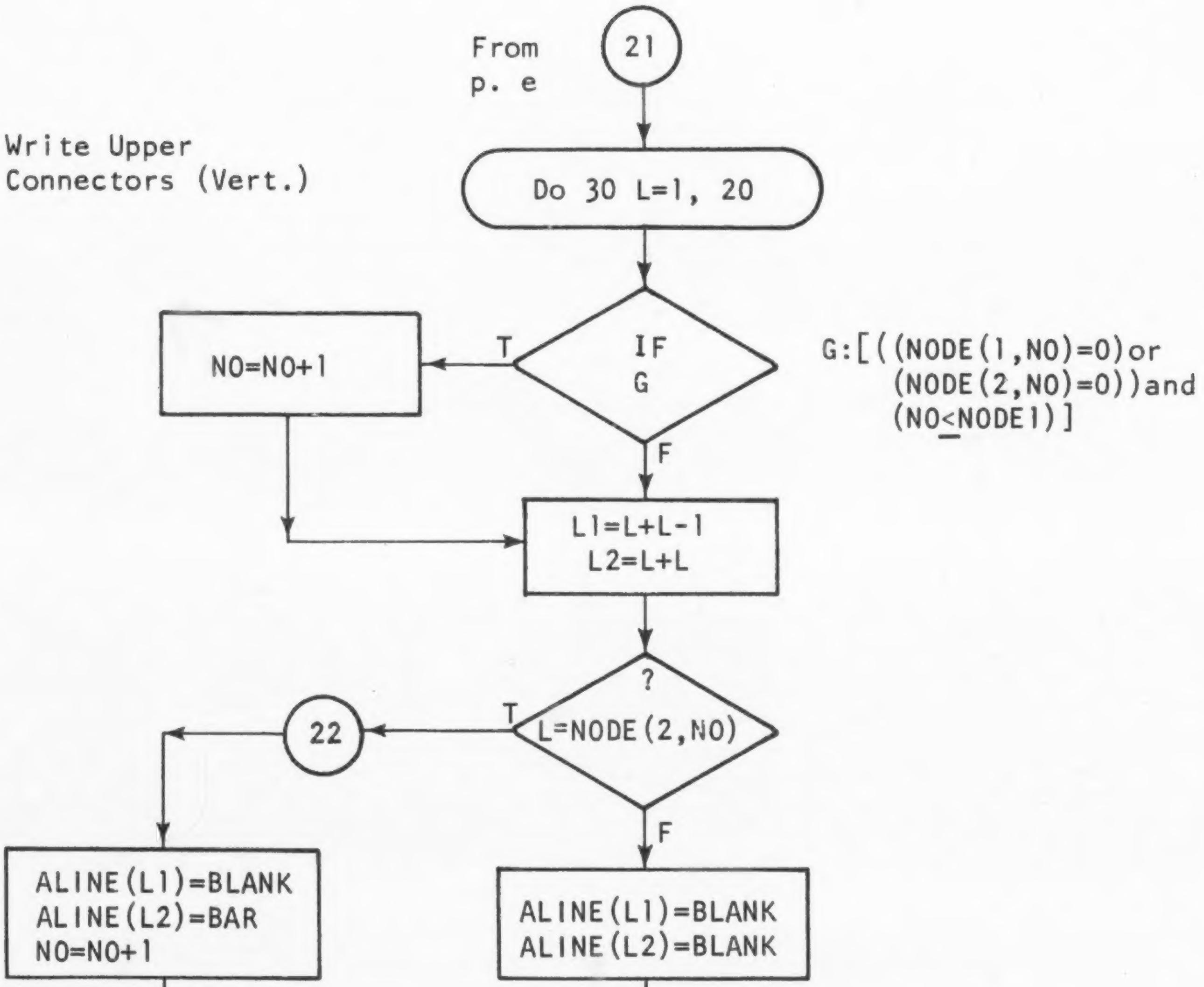
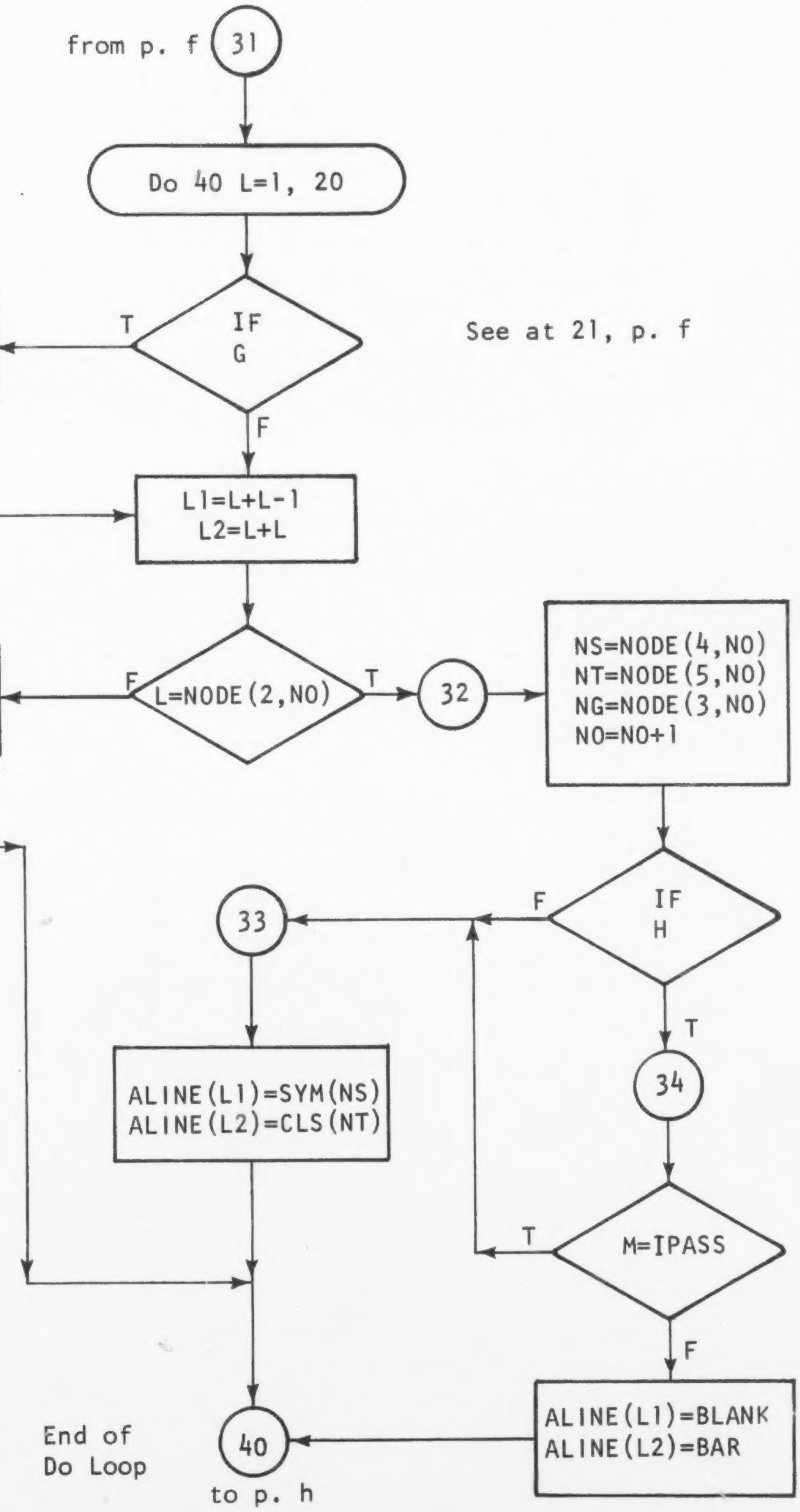

From
p. e
21
Write Upper
Connectors (Vert.)
Do 30 L=1, 20
IF
G
T
F
NO=NO+1
G:[((NODE(1,NO)=0)or
(NODE(2,NO)=0))and
(NO≤NODE1)]
L1=L+L-1
L2=L+L
?
L=NODE(2,NO)
T
22
F
ALINE(L1)=BLANK
ALINE(L2)=BAR
NO=NO+1
ALINE(L1)=BLANK
ALINE(L2)=BLANK
30
end of Do Loop
NO=NO-ITREE(M)
PRINT ALINE
IUPLOW=2
T
50
F
31
to p. g

Figure 3.7f Subprogram DIAGRM

(4) Write Line of Nodes


from p. f
31
Do 40 L=1, 20
IF G
T
F
NO=NO+1
See at 21, p. f
L1=L+L-1
L2=L+L
L=NODE(2,NO)
F
T
ALINE(L1)=BLANK
ALINE(L2)=BLANK
32
NS=NODE(4,NO)
NT=NODE(5,NO)
NG=NODE(3,NO)
NO=NO+1
IF H
F
T
33
34
ALINE(L1)=SYM(NS)
ALINE(L2)=CLS(NT)
M=IPASS
T
F
ALINE(L1)=BLANK
ALINE(L2)=BAR
40
End of Do Loop
to p. h

H: [(NS=NODE(4,NG)) and NT=NODE(5,NG)]

Figure 3.7g Subprogram DIAGRM

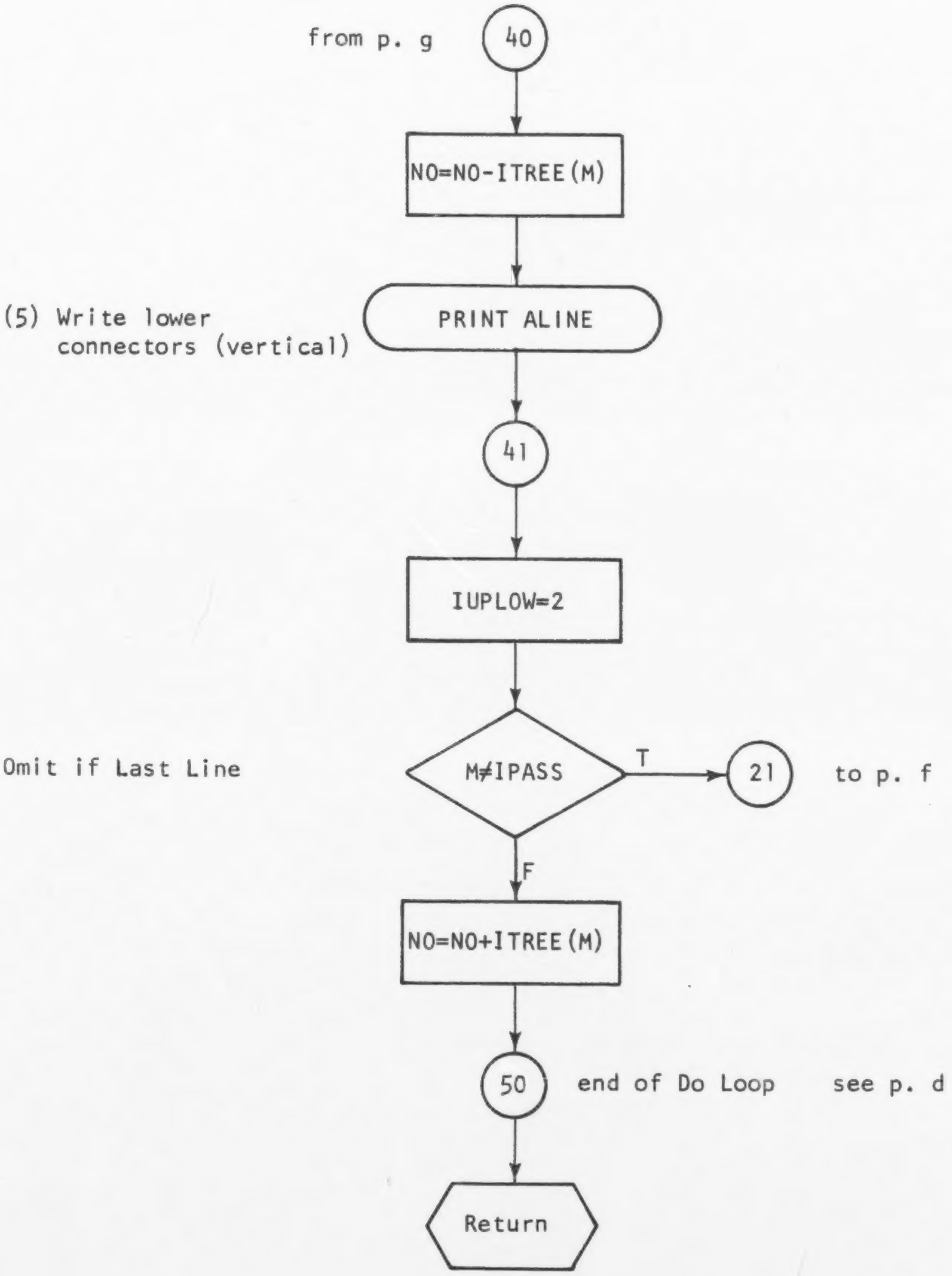

from p. g
40
NO=NO-ITREE(M)
(5) Write lower connectors (vertical)
PRINT ALINE
41
IUPLOW=2
Omit if Last Line
M≠IPASS
T
21
to p. f
F
NO=NO+ITREE(M)
50
end of Do Loop
see p. d
Return

Figure 3.7h Subprogram DIAGRM

the "present rule symbol" is not a variable symbol, then flag an error by setting MATCH to "false".

(7) For each subscript of the "present rule symbol" perform the appropriate tests in steps 8 through 13.

(8) If the subscript is not a dependent variable, skip to step 11.

(9) If the subscript is a Root variable and if the "present rule symbol" is not a variable symbol, skip to step 13.

(10) If the variable storage register is empty for this subscript (VARSTR(M)=0), then store the inputted value of this subscript in the variable storage register, and in either case, set the value of this subscript for the "present rule symbol" equal to the value stored in the variable storage register. This step is used for compounding symbols; skip to step 13.

(11) If the present subscript is an independent variable (SYMBOL(M)=9) and either the "present rule symbol" is not compounded (INREAD(4)=0) or the present subscript is related to compounding ($M \leqq 6$), then set the value of this subscript for the "present rule symbol" equal to the input value previously read in.

(12) If the value of this subscript for the "present rule symbol" is -1, then set the value of this subscript for the "present rule symbol" equal to the stored value diminished by 1 (SYMBOL(M) = VARSTR(M)-1). This is used for compounding symbols.

(13) If this is not the last subscript, go to the next subscript and return to step 8, otherwise continue to step 14.

(14) Set the values of subscripts 24 and 29 for the "present rule symbol" equal to value of the corresponding input subscripts previously read in.

(15) If the "present rule symbol" is a variable symbol, set its symbol type equal to that

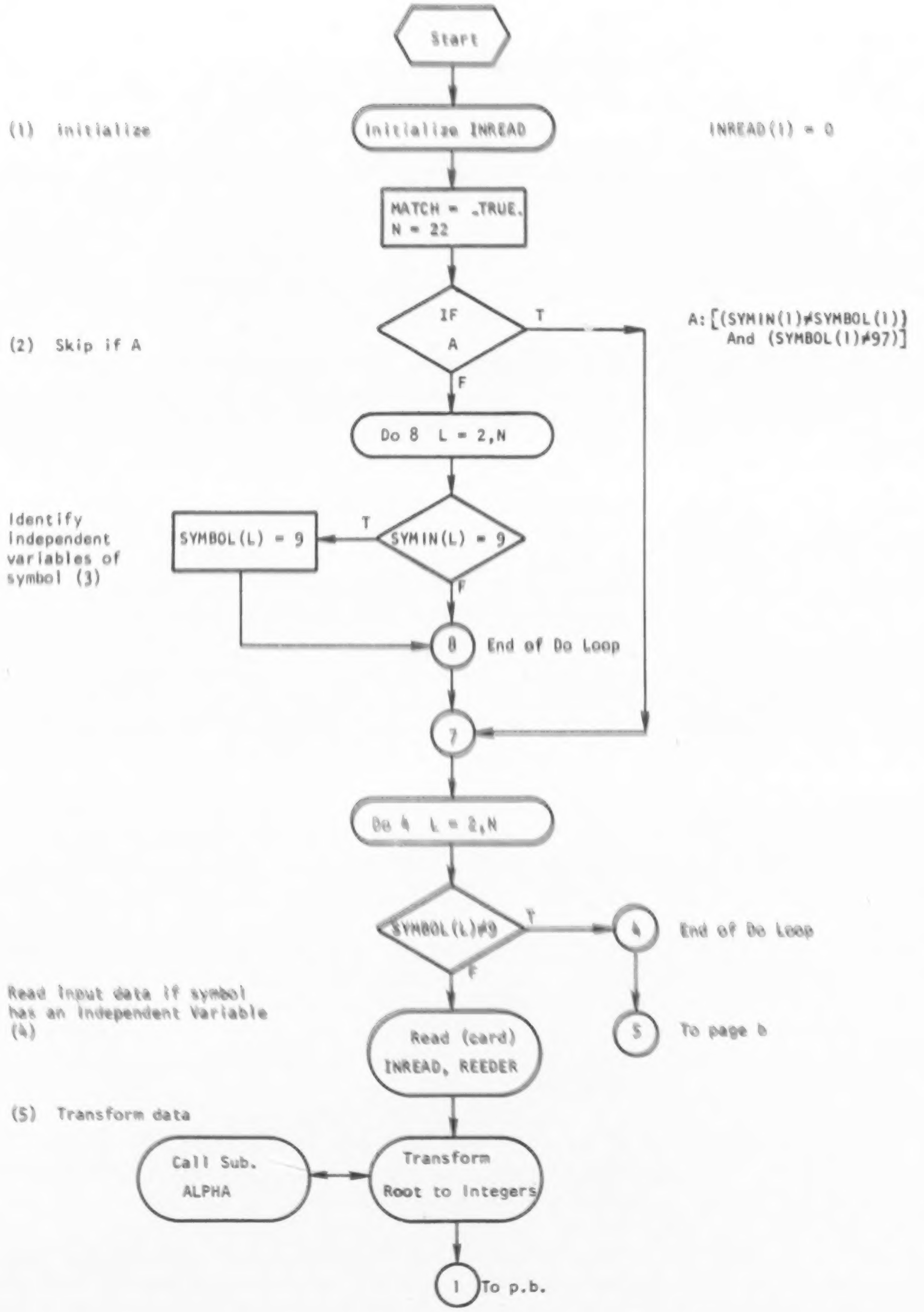

Start
(1) initialize
Initialize INREAD
INREAD(1) = 0
MATCH = .TRUE.
N = 22
(2) Skip if A
IF
A
T
F
A: [(SYMIN(1)≠SYMBOL(1))
And (SYMBOL(1)≠97)]
Do 8 L = 2,N
Identify
independent
variables of
symbol (3)
SYMBOL(L) = 9
T
SYMIN(L) = 9
F
8
End of Do Loop
7
Do 4 L = 2,N
SYMBOL(L)≠9
T
F
4
End of Do Loop
5
To page b
Read input data if symbol
has an independent Variable
(4)
Read (card)
INREAD, REEDER
(5) Transform data
Call Sub.
ALPHA
Transform
Root to Integers
1
To p.b.

Figure 3.8a Subprogram INPUT

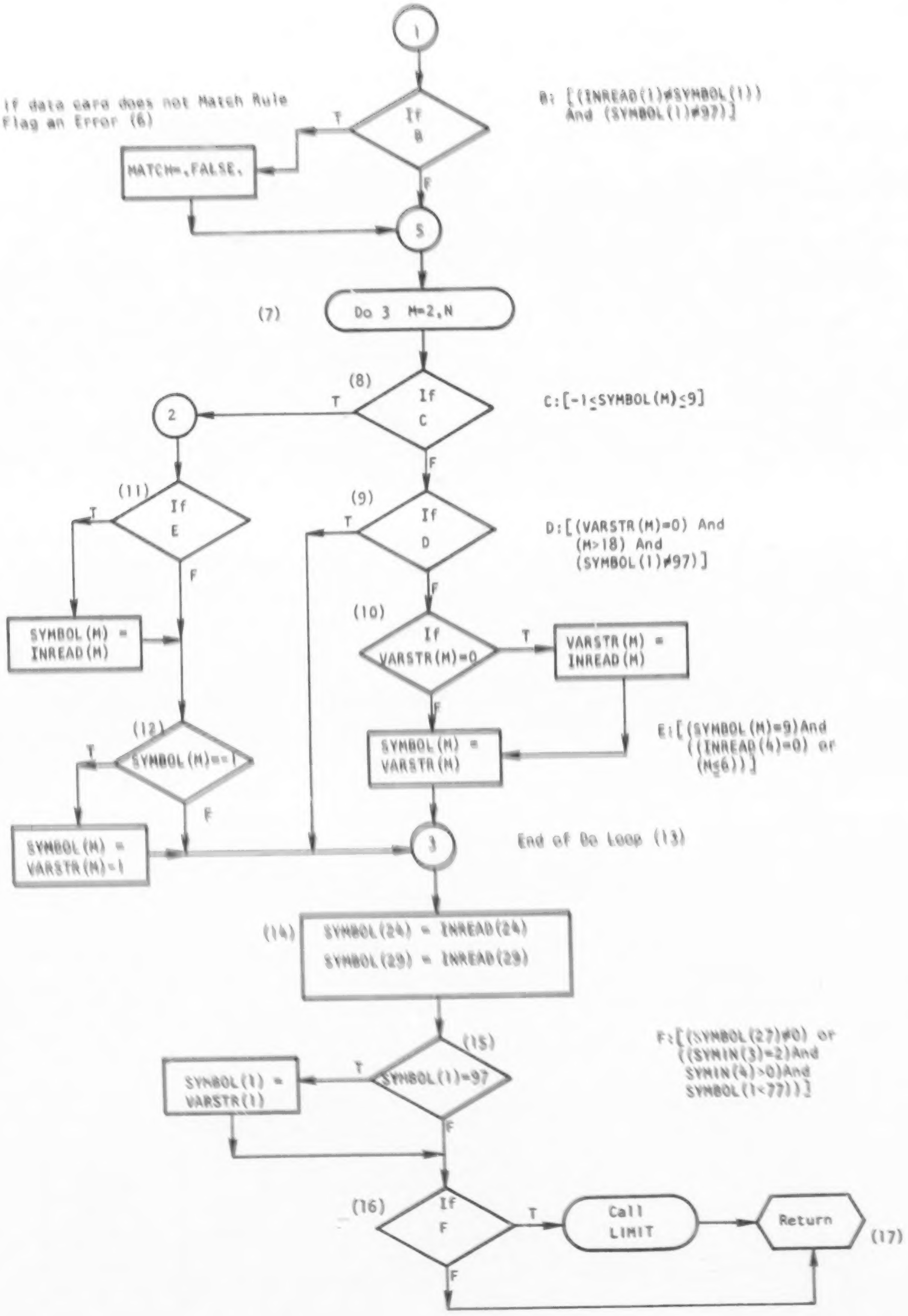

1
If data card does not Match Rule
Flag an Error (6)
If
B
T
F
MATCH=.FALSE.
B: [(INREAD(1)≠SYMBOL(1))
And (SYMBOL(1)≠97)]
5
(7)
Do 3 M=2,N
(8)
If
C
C:[-1≤SYMBOL(M)≤9]
2
(11)
If
E
(9)
If
D
D:[(VARSTR(M)=0) And
(M>18) And
(SYMBOL(1)≠97)]
SYMBOL(M) =
INREAD(M)
(10)
If
VARSTR(M)=0
VARSTR(M) =
INREAD(M)
(12)
SYMBOL(M)=-1
SYMBOL(M) =
VARSTR(M)
E:[(SYMBOL(M)=9)And
((INREAD(4)=0) or
(M≤6))]
SYMBOL(M) =
VARSTR(M)-1
3
End of Do Loop (13)
(14)
SYMBOL(24) = INREAD(24)
SYMBOL(29) = INREAD(29)
(15)
SYMBOL(1) =
VARSTR(1)
SYMBOL(1)=97
F:[(SYMBOL(27)≠0) or
((SYMIN(3)=2)And
SYMIN(4)>0)And
SYMBOL(1<77))]
(16)
If
F
Call
LIMIT
Return
(17)

Figure 3.8b Subprogram INPUT

of the symbol stored in the variable storage register. This is used for compounding symbols.

(16) If there are limitations set on the "present rule symbol" see if these limitations are met by using Subprogram LIMIT.

(17) When all specified operations are completed, return to the calling program.

3.3.1.7 Subprogram LIMIT

This subprogram (Figure 3.9) is called by Subprogram RULRIT and by Subprogram INPUT to test symbols for certain specified limitations on the values of their subscripts. The limitations are specified by Subscripts 27 and 28. Subscript 27 specifies the restraint type and Subscript 28 specifies the subscript number to which the restraint applies.

The restraint type refers to a row number IR in Restraint Matrix RESTRT(L,IR). The first number in this row, RESTRT(1,IR), tells how many restraint numbers are on the list; the remaining numbers in this row are limitations on the specified subscript. The logic of the limitations is as follows: the value of the specifies subscript must be equal to one of the positive numbers on the list, or it must not be equal to the absolute value of the negative number on the list. If these restraints are not met the subprogram sets the logic monitor MATCH to "false."

In addition the subprogram is used to check the permissible permutations of number, gender and person for symbols compounded after class 2. The logic follows the following table:[10]

SYMIN					SYMBOL			MATCH
F	B	N	G	P	N	G	P	
2	-	-	2	-	-	≠2	-	FALSE
2	-	-	-	2	-	-	1	FALSE
2	-	-	-	3	-	-	≠3	FALSE
2	2	1	-	-	≠1	-	-	FALSE
Otherwise								TRUE

---

[10] These computations are dependent on the "content" of the grammar.

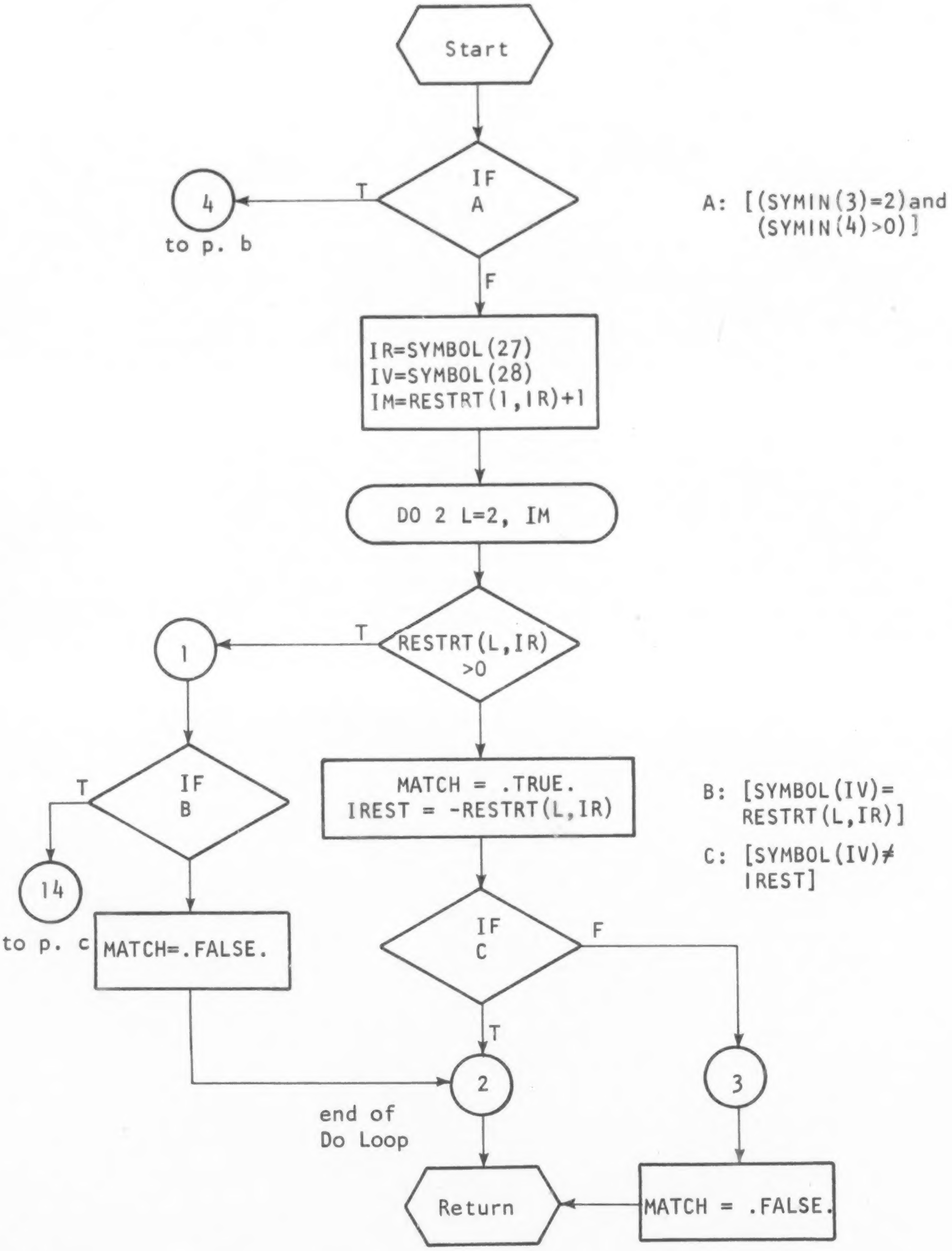

Start
IF A
T
4
to p. b
F
A: [(SYMIN(3)=2)and (SYMIN(4)>0)]
IR=SYMBOL(27)
IV=SYMBOL(28)
IM=RESTRT(1,IR)+1
DO 2 L=2, IM
RESTRT(L,IR) >0
T
1
IF B
T
14
to p. c
MATCH=.FALSE.
MATCH = .TRUE.
IREST = -RESTRT(L,IR)
B: [SYMBOL(IV)= RESTRT(L,IR)]
C: [SYMBOL(IV)≠ IREST]
IF C
F
T
2
end of Do Loop
3
Return
MATCH = .FALSE.

Figure 3.9a Subprogram LIMIT

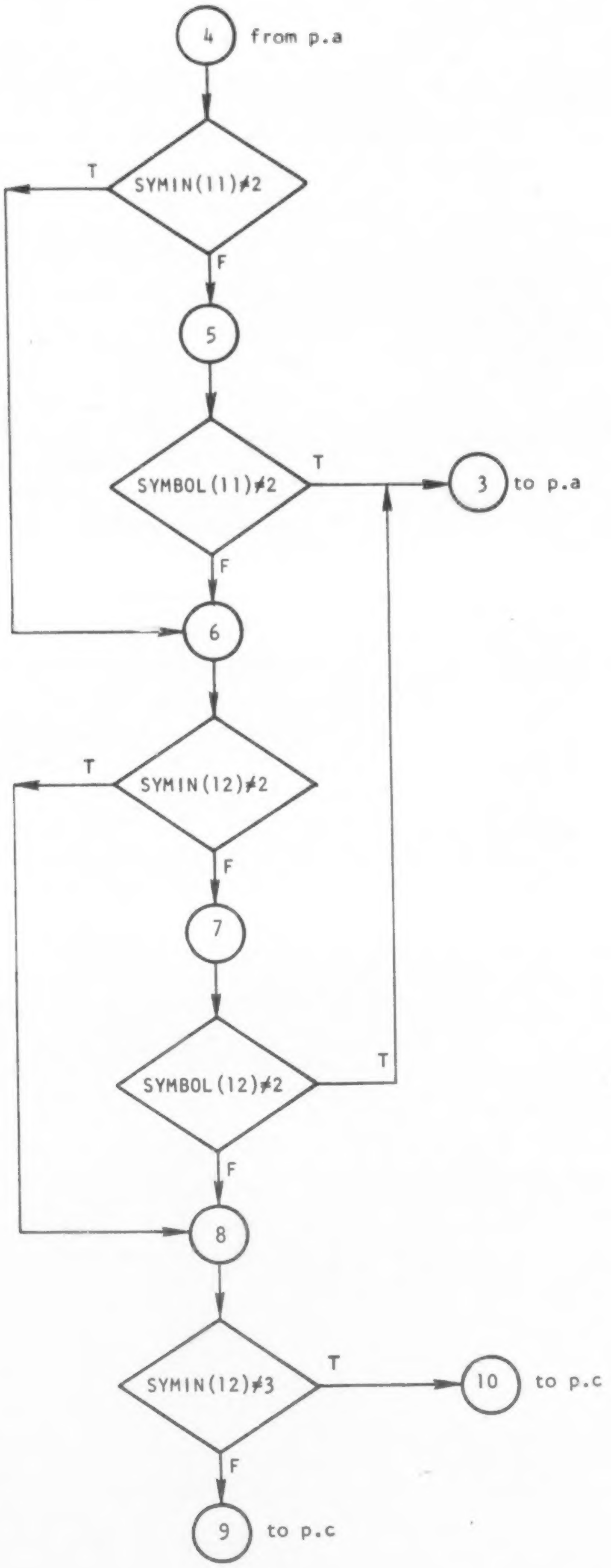

4
from p.a
SYMIN(11)≠2
T
F
5
SYMBOL(11)≠2
T
F
3
to p.a
6
SYMIN(12)≠2
T
F
7
SYMBOL(12)≠2
T
F
8
SYMIN(12)≠3
T
F
10
to p.c
9
to p.c

Figure 3.9b Subprogram LIMIT

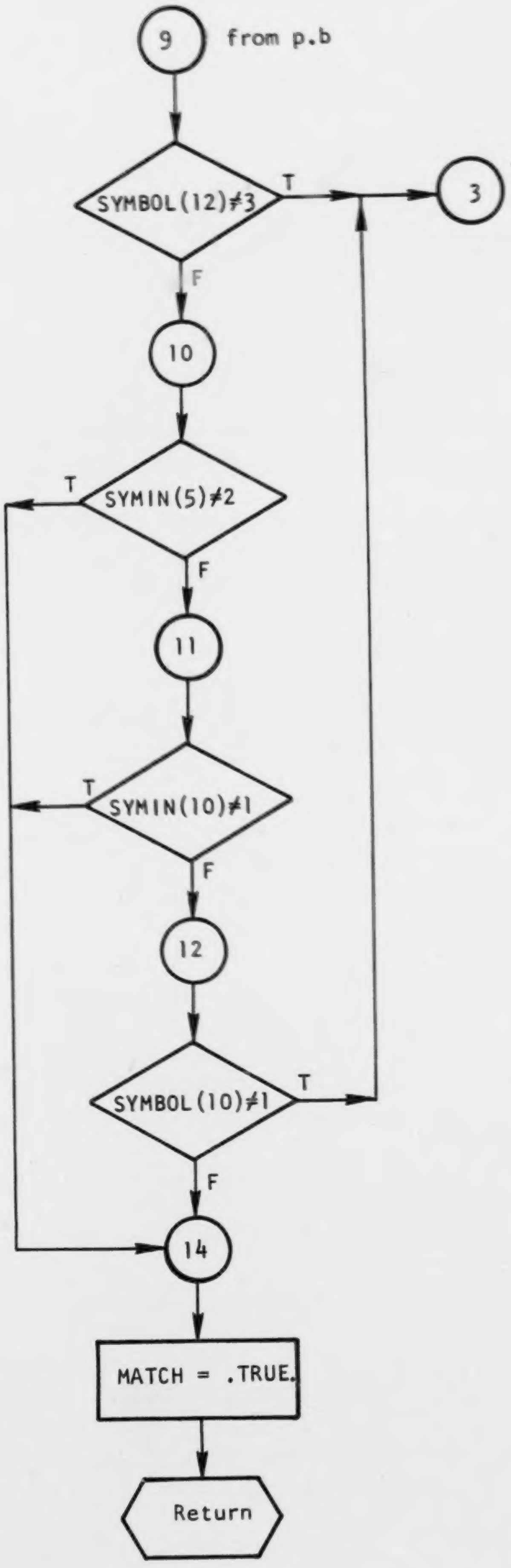

9
from p.b
SYMBOL(12)≠3
T
F
3
10
SYMIN(5)≠2
T
F
11
SYMIN(10)≠1
T
F
12
SYMBOL(10)≠1
T
F
14
MATCH = .TRUE.
Return

Figure 3.9c Subprogram LIMIT

3.3.1.8 Subprogram ORTHOG

This subprogram (Figure 3.10) is called by the Main Program SENSYN (step 20) to compute the correct spelling of each word in the generated sentence (using Subprogram INFLEX), to edit the resultant sequence of symbols by removing Word Connectors and by inserting spaces in the correct places, and to print out the edited sentence on the high speed printer.

The subprogram performs the following operations:

(1) Initializes index IS and editing register ALINE to erase previous data (Figure 3.10a).

(2) For each of the ITERM symbols in the terminal string of symbols performs steps 3 through 12 (Figure 3.10a).

(3) Transfers the present (M-th) symbol from the list of terminal symbols (ITABLE) to the "present constituent" register (SYMBOL) (Figure 3.10a).

(4) Computes the correct spelling for the symbol in the "present constituent" register by using Subprogram INFLEX, which returns the inflected word in register WORD (Figure 3.10a).

(5) Checks to see if editing register ALINE is full, if not, it skips to step 10 (Figure 3.10a).

(6) Edits data in the editing register (Figure 3.10b) to remove Word Connectors and insert spaces according to the following criteria:

→ Space

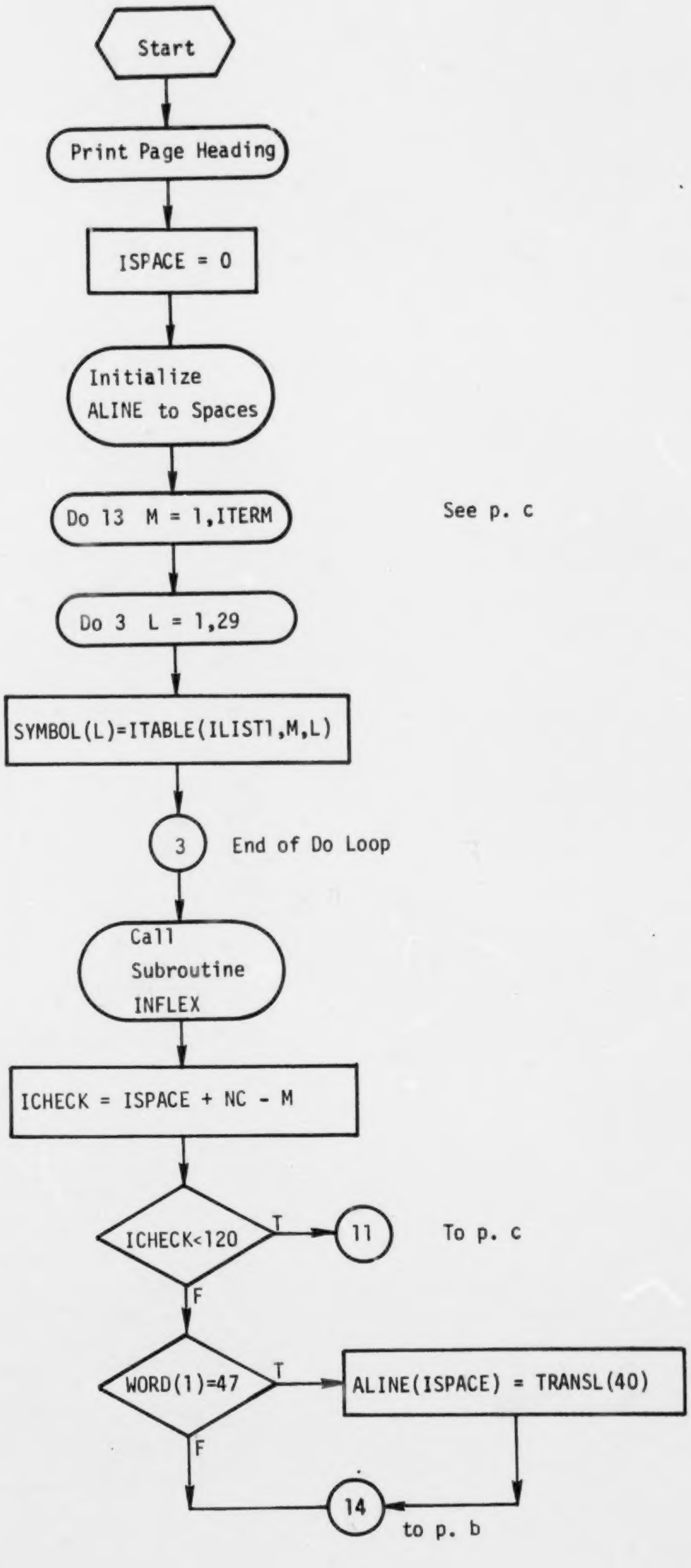
Δ → No Space

Δ # → No Space

(7) Fills the unused portion of the editing register with spaces (Figure 3.10b).

(8) Prints out the line of edited text from the editing register (Figure 3.10c).

(9) Initializes indes IS and the editing register ALINE to erase above data (Figure 3.10c).


Start
Print Page Heading
ISPACE = 0
Initialize ALINE to Spaces
Do 13 M = 1,ITERM
See p. c
Do 3 L = 1,29
SYMBOL(L)=ITABLE(ILIST1,M,L)
3
End of Do Loop
Call Subroutine INFLEX
ICHECK = ISPACE + NC - M
ICHECK<120
T
11
To p. c
F
WORD(1)=47
T
ALINE(ISPACE) = TRANSL(40)
F
14
to p. b

Erase Previous Line from memory (1)

Compute Orthography of ITERM words. (2)

Store Mth terminal symbol in SYMBOL and compute orthog. (3)

Computes orthog. of Word defined by symbol & returns correct orthog. in WORD. (4)

Check for end of line if so, print sentence up to this point (5)

Figure 3.10a Subprogram ORTHOG

1960

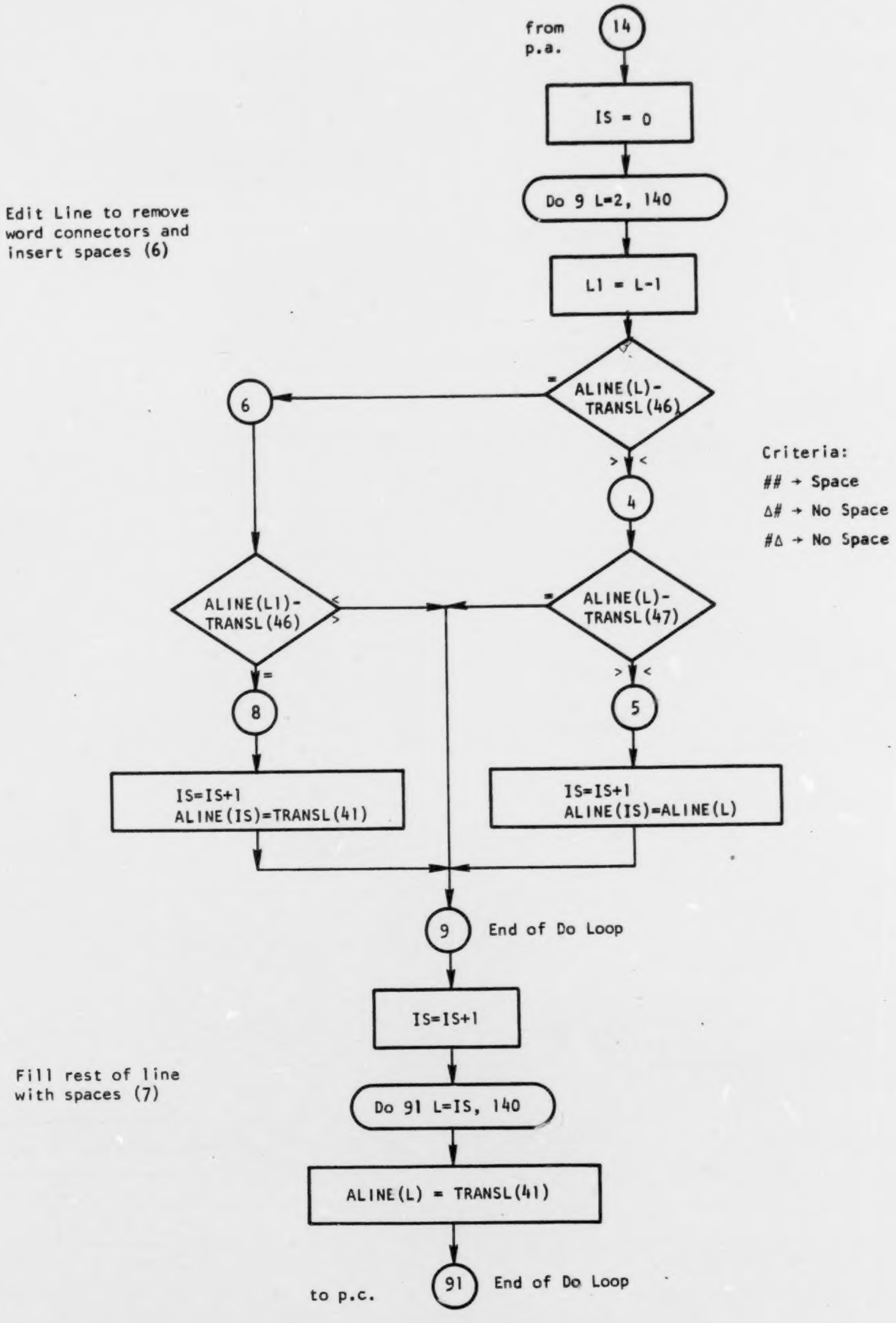

from p.a.
14
IS = 0
Do 9 L=2, 140
Edit Line to remove word connectors and insert spaces (6)
L1 = L-1
ALINE(L)-TRANSL(46)
=
6
> <
4
Criteria:
## → Space
Δ# → No Space
#Δ → No Space
ALINE(L1)-TRANSL(46)
<
>
=
ALINE(L)-TRANSL(47)
=
> <
8
5
IS=IS+1
ALINE(IS)=TRANSL(41)
IS=IS+1
ALINE(IS)=ALINE(L)
9
End of Do Loop
IS=IS+1
Fill rest of line with spaces (7)
Do 91 L=IS, 140
ALINE(L) = TRANSL(41)
to p.c.
91
End of Do Loop

Figure 3.10b Subprogram ORTHOG

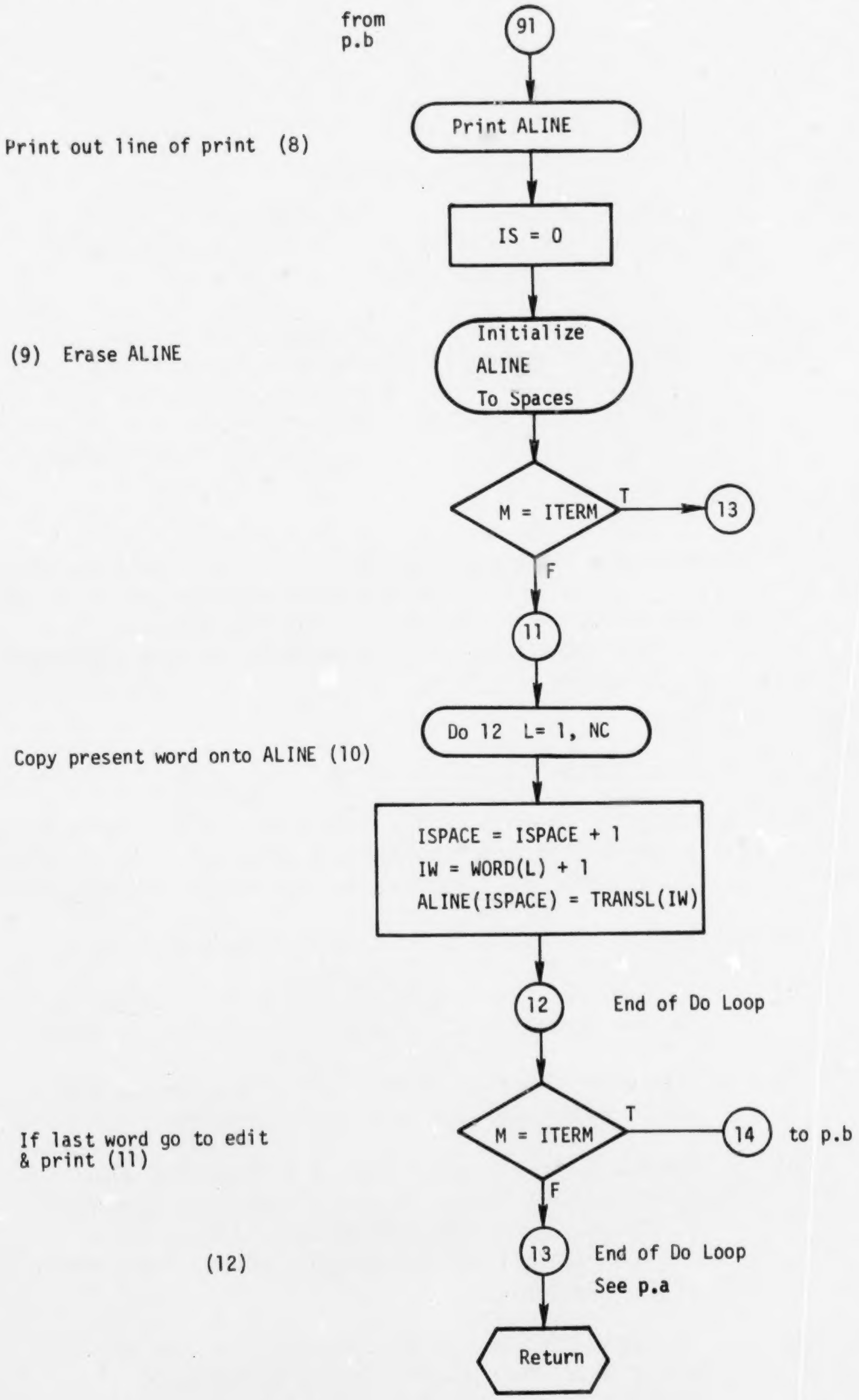

from
p.b
91
Print ALINE
Print out line of print (8)
IS = 0
Initialize
ALINE
To Spaces
(9) Erase ALINE
M = ITERM
T
13
F
11
Do 12 L= 1, NC
Copy present word onto ALINE (10)
ISPACE = ISPACE + 1
IW = WORD(L) + 1
ALINE(ISPACE) = TRANSL(IW)
12
End of Do Loop
If last word go to edit
& print (11)
M = ITERM
T
14
to p.b
F
(12)
13
End of Do Loop
See p.a
Return

Figure 3.10c Subprogram ORTHOG

(10) Copies present inflected word onto the next available spaces of the editing register (Figure 3.10c).

(11) If present symbol is the last one of the string, skips back to step 6, otherwise continue (Figure 3.10c).

(12) If present symbol is not the last of the string, increments the symbol index M and skips back to step 3, otherwise returns to the calling program (Figure 3.10c).

3.3.1.9 Subprogram OUTPUT

This subprogram (Figure 3.11) is called by the Main Program SENSYN (steps 10 and 15) to print out the data associated with each symbol in the string after each rerite pass of the grammar on a string of symbols. See Appendix C for an example of the resultant output from this subprogram.

3.3.1.10 Subprogram RERITE

This subprogram (Figure 3.12) is called by the Main Program SENSYN (step 13) to apply the appropriate replacement rule to a given symbol and write the replacement symbols on the rerite symbol list.

The subprogram performs the following operations:

(1) Places the Ith symbol of the list of symbols in the "present constituent" register (SYMIN).

(2) Computes the rule number that applies to the given symbol using Subprogram RULENO.

(3) Checks to see that there is a rule for the given symbol if it is not a terminal symbol; if there is no rule, it prints out an error message and stops the computations, otherwise it continues.

(4) It computes the correct values for the subscripts of the symbols to be written on the rewrite list, and using Subprogram RULRIT, writes these symbols on the rewrite list.

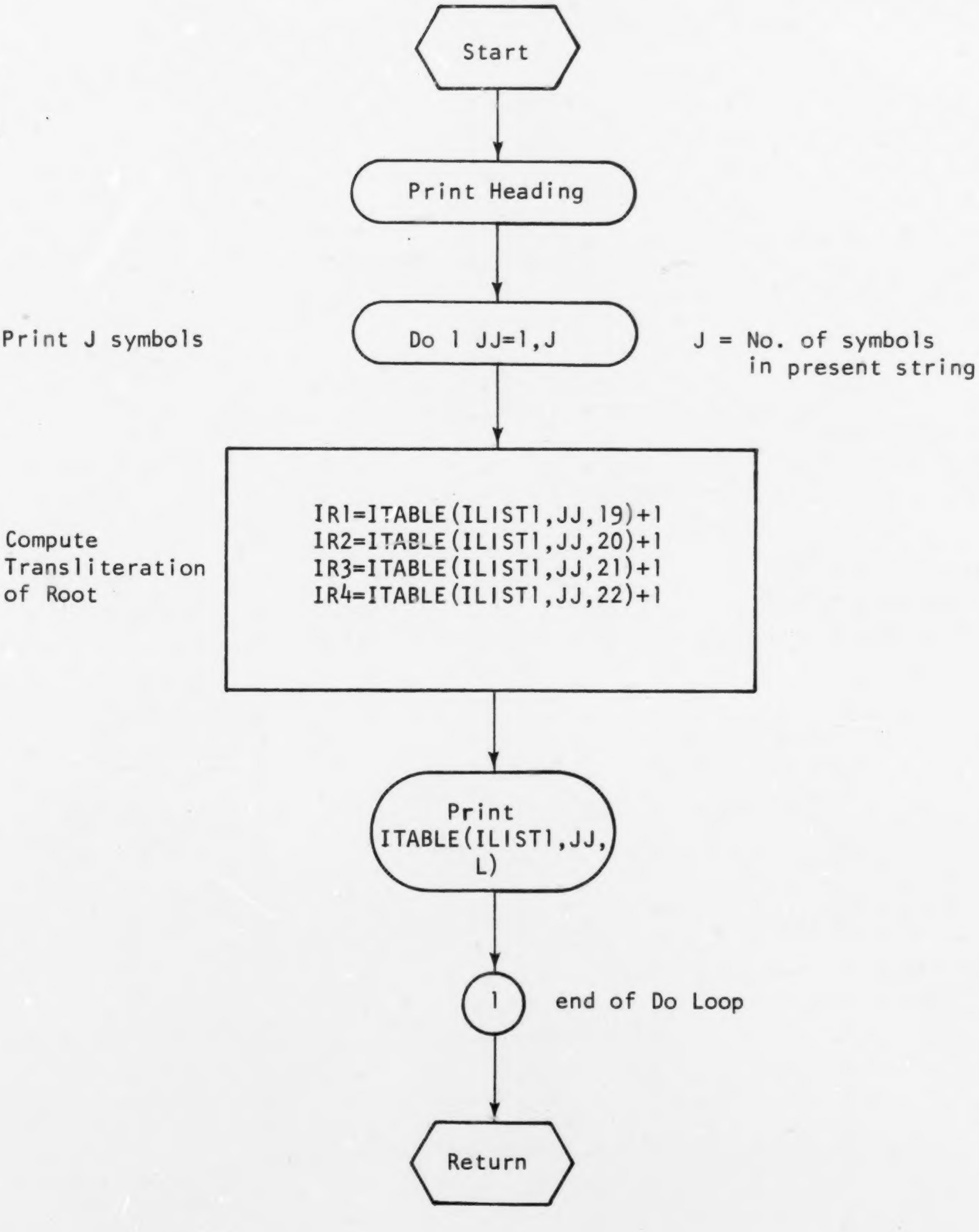
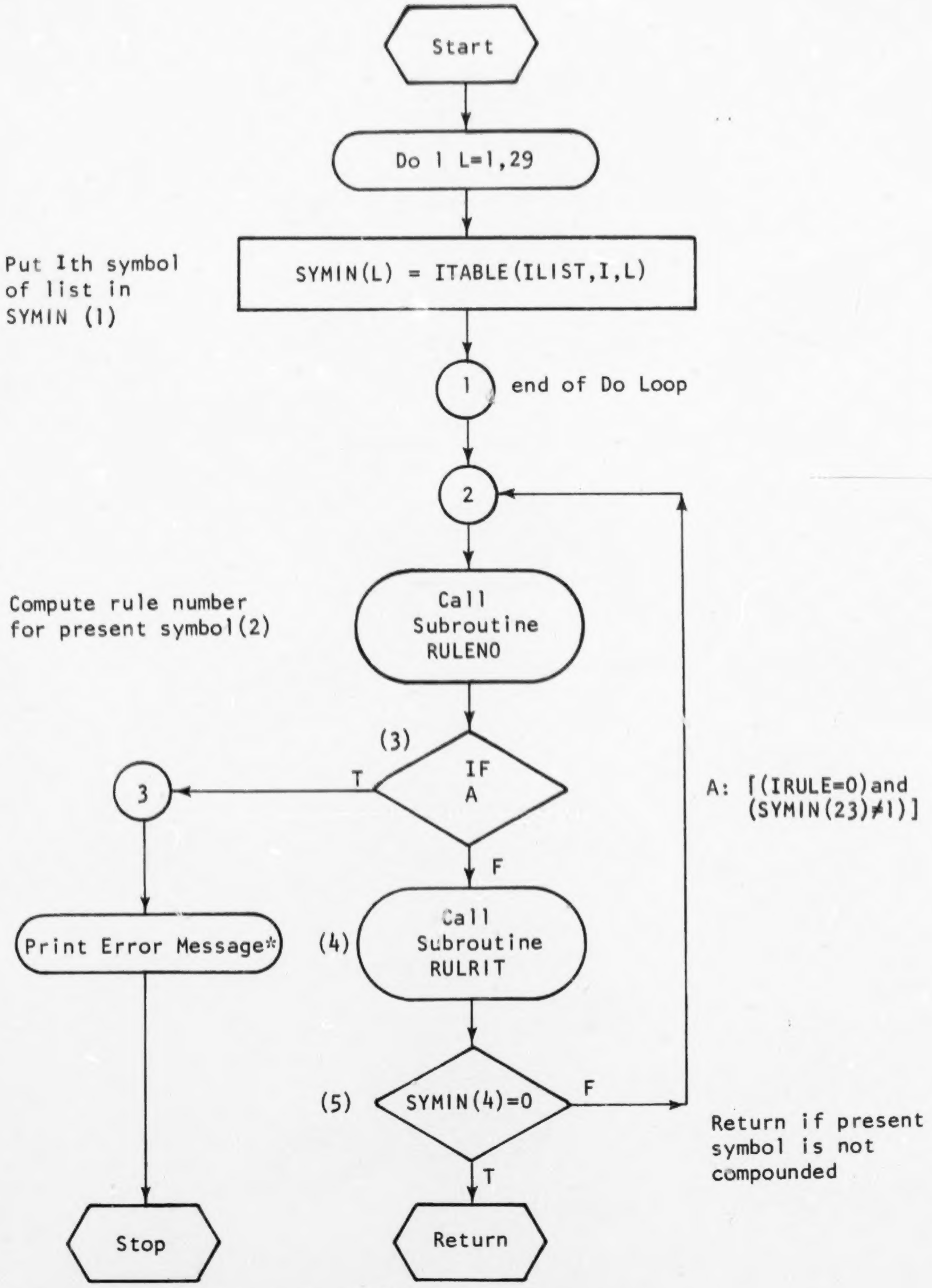

Start
Print Heading
Print J symbols
Do 1 JJ=1,J
J = No. of symbols in present string
Compute Transliteration of Root
IR1=ITABLE(ILIST1,JJ,19)+1
IR2=ITABLE(ILIST1,JJ,20)+1
IR3=ITABLE(ILIST1,JJ,21)+1
IR4=ITABLE(ILIST1,JJ,22)+1
Print ITABLE(ILIST1,JJ,L)
1
end of Do Loop
Return

Figure 3.11 Subprogram OUTPUT


Start
Do 1 L=1,29
Put Ith symbol of list in SYMIN (1)
SYMIN(L) = ITABLE(ILIST,I,L)
1
end of Do Loop
2
Compute rule number for present symbol(2)
Call Subroutine RULENO
(3)
IF A
T
F
3
A: [(IRULE=0)and (SYMIN(23)≠1)]
Print Error Message*
(4)
Call Subroutine RULRIT
(5)
SYMIN(4)=0
F
T
Return if present symbol is not compounded
Stop
Return

*No rule number found for symbol ______.

Figure 3.12 Subprogram RERITE

(5) It repeats from step 2 if the present symbol in SYMIN is compounded, otherwise it returns to the calling program.

3.3.1.11 Subprogram RULENO

This subprogram (Figure 3.13) is called by Subprogram RERITE (step 2) to compute the grammar rule number that applies to a given symbol. The grammar rules are stored in Matrix RULE, and the computed rule numbers are as follows:

IRULE = the number of the first element of the first rule that applies to the given symbol

IMAX = the number of the last element of the last rule that applies to a given symbol.

For simple, non-negated symbols the applicable rule numbers initially were computed in Main Program SENSYN (step 4) and stored in rule catalogue IPSI. The data is stored so that for Symbol IS, Class IC, IRULE = IPSI(IS,IC,1) and IMAX = IPSI(IS,IC,2).

The subprogram computes the rule numbers according to the following hierarchy:

(a) Compound symbols

(b) Negated symbols

(c) Simple, non-negated symbols.

The following operations are performed:

(1) If the given symbol is a compounded symbol, skips to step 4.

(2) If the symbol is a negated symbol, skips to step 5.

(3) For simple, non-negated symbols, computes IS, and IC, and IRULE and IMAX from rule catalogue IPSI, and returns.

(4) For compounded symbols, computes the values of IRULE and IMAX based on the value of the compounding class NF, and returns.[11]

---

[11] These computations are dependent on the "content" of the grammar.

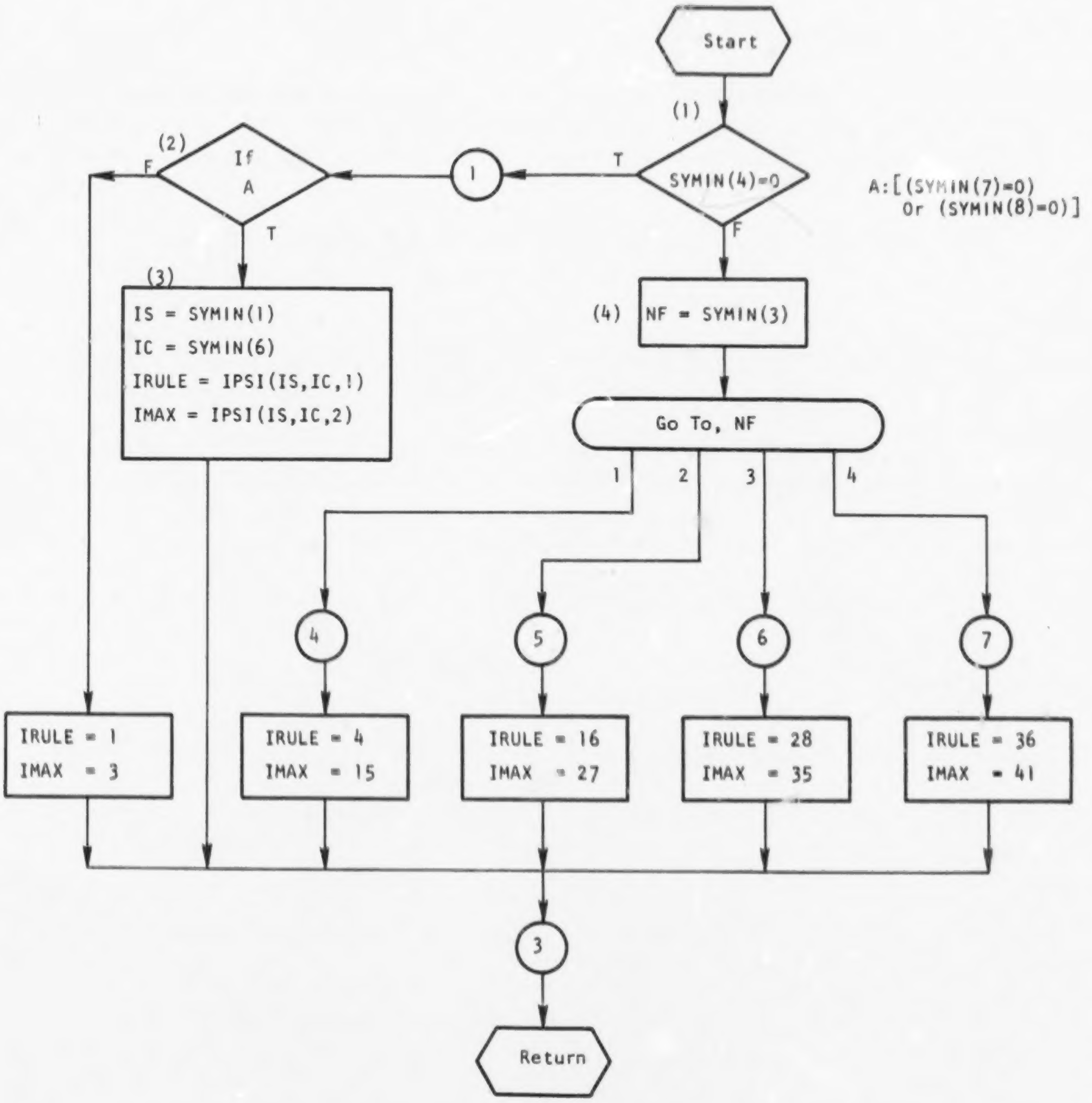

Start
(1)
SYMIN(4)=0
T
F
A:[(SYMIN(7)=0) Or (SYMIN(8)=0)]
1
(2)
If A
F
T
(3)
IS = SYMIN(1)
IC = SYMIN(6)
IRULE = IPSI(IS,IC,1)
IMAX = IPSI(IS,IC,2)
(4)
NF = SYMIN(3)
Go To, NF
1
2
3
4
4
5
6
7
IRULE = 1
IMAX = 3
IRULE = 4
IMAX = 15
IRULE = 16
IMAX = 27
IRULE = 28
IMAX = 35
IRULE = 36
IMAX = 41
3
Return

Figure 3.13 Subprogram RULENO

(5) For negated symbols, computes the value IRULE = 1 and IMAX = 3, and returns.[11]

3.3.1.12 Subprogram RULRIT

This subprogram (Figure 3.14) is called by Subprogram RERITE (step 4) to replace a given symbol in the string of symbols with one or more symbols on the rewrite list of symbols in accordance with the given grammar rule and computed subscript values. The following operations are performed:

(1) If the symbol is a simple, non-negated terminal symbol, skip to step 21 (Figure 3.14a).

(2) Put the first symbol of the given rule in the "present rule symbol" register SYMBOL.

(3) Check to see that the "present rule symbol" matches the "present constituent" with respect to the criteria of the grammar, using Subprogram SYMACH.

(4) If the symbols match (MATCH = True) compute the value of the various dependent variable subscripts for the "present rule symbol" using Subprogram VARATT (Figure 3.14a).

(5) Check the symbols for meeting the requirements of any restraints specified by the rules, if they fail, set MATCH TO "false" (Figure 3.14a).

(6) Skip to step 20 if symbols do not match (MATCH = "false") (Figure 3.14a).

(7) Compute certain indices to govern progress in replacement of symbols (Figure 3.14b).

(8) For each symbol on the right side of the given grammar rule perform steps 9 through 18 (Figure 3.14b).

(9) Put the K-th element of the right hand side of the rule in the "present rule symbol" register SYMBOL (Figure 3.14b).

(10) If the "present constituent" is a terminal symbol and the "present rule symbol" is not a negative mark the "present rule symbol" as terminal (Figure 3.14b).

(11) If the "present rule symbol" is part of a compounding rule requiring the reduction of subscript 4, reduce subscript 4 of the "present constituent" by 1 and return to the calling program, otherwise continue (Figure 3.14b).

(12) Compute the values of the independent variable subscripts for the "present rule symbol" using Subprogram INPUT (Figure 3.14b).

(13) If the computed values met all the restraint requirements (MATCH = "true"), continue, otherwise print out an error message and stop computation (Figure 3.14b).

(14) If the "present rule symbol" is optional and the input data "excludes" it, skip to step 18 after adjusting subscript 26 (Figure 3.14c).

(15) If the "present rule symbol" is mandatory as specified by the associated rule, skip to step 17 (Figure 3.14c).

(16) Write an error message regarding the lack of agreement of data for subscript 2 (Figure 3.14c).

(17) Write "present rule symbol" on the new list of symbols and adjust certain indices (Figure 3.14c).

(18) If the "present rule symbol" is not the last symbol of the given rule, advance index K and return back to step 9 (Figure 3.14c), otherwise continue.

(19) If the "present constituent" is compounded and the value of subscript 4 is equal to or less than 2, set it equal to zero, and return to the calling program.

(20) Move to the next rule by advancing IRULE to 1 + IEND. If there is another rule that applies to the "present consituent" (IRULE < IMAX) return to step 2, otherwise write and error message indicating that no rule was found that applies to the "present constituent", and stop computation (Figure 3.14d).

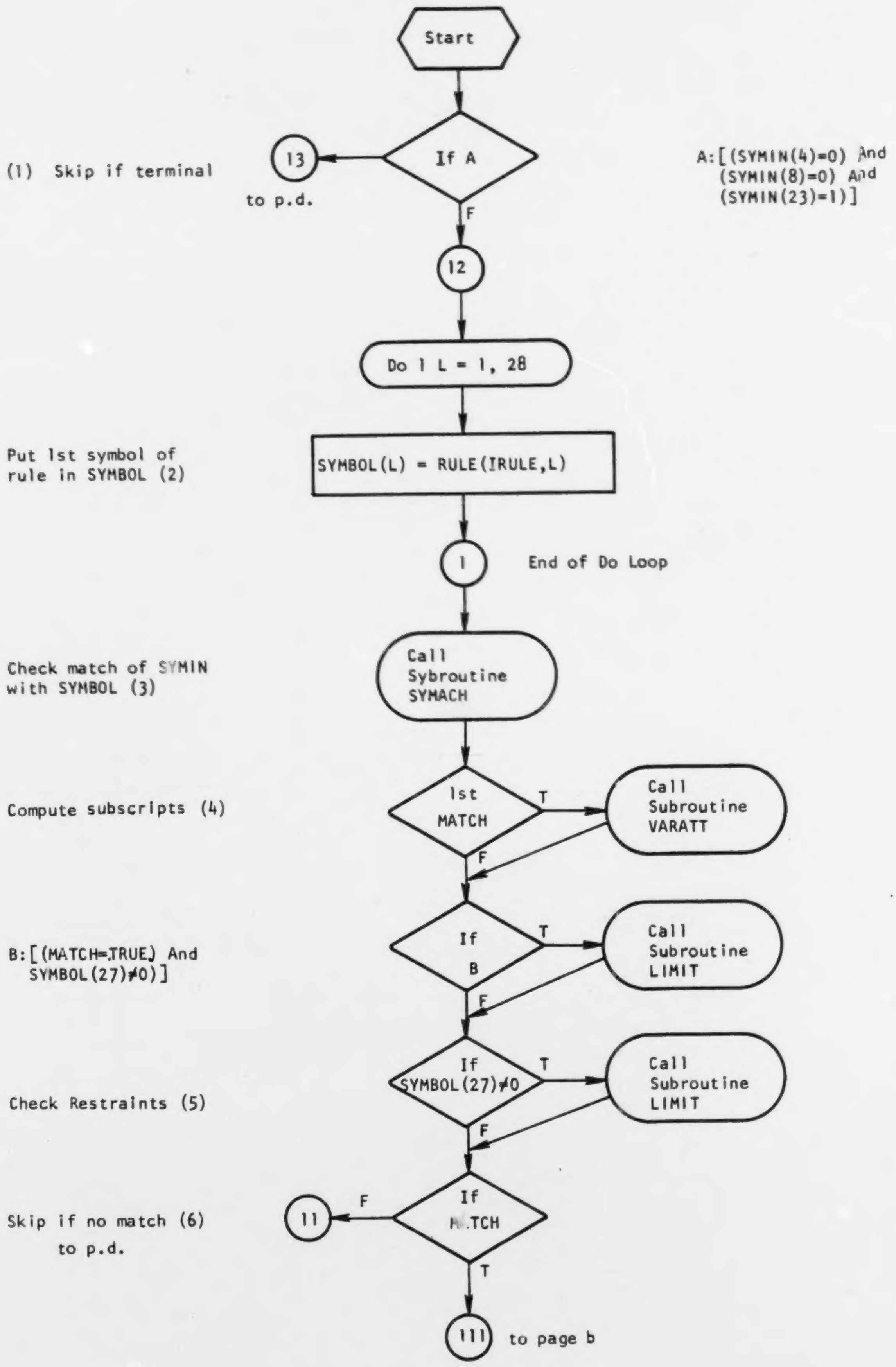

Start
(1) Skip if terminal
13
to p.d.
If A
A:[(SYMIN(4)=0) And
(SYMIN(8)=0) And
(SYMIN(23)=1)]
F
12
Do 1 L = 1, 28
Put 1st symbol of
rule in SYMBOL (2)
SYMBOL(L) = RULE(IRULE,L)
1
End of Do Loop
Check match of SYMIN
with SYMBOL (3)
Call
Sybroutine
SYMACH
Compute subscripts (4)
1st
MATCH
T
F
Call
Subroutine
VARATT
B:[(MATCH=TRUE) And
SYMBOL(27)≠0)]
If
B
T
F
Call
Subroutine
LIMIT
Check Restraints (5)
If
SYMBOL(27)≠0
T
F
Call
Subroutine
LIMIT
Skip if no match (6)
to p.d.
11
F
If
MATCH
T
111
to page b

Figure 3.14a Subprogram RULRIT

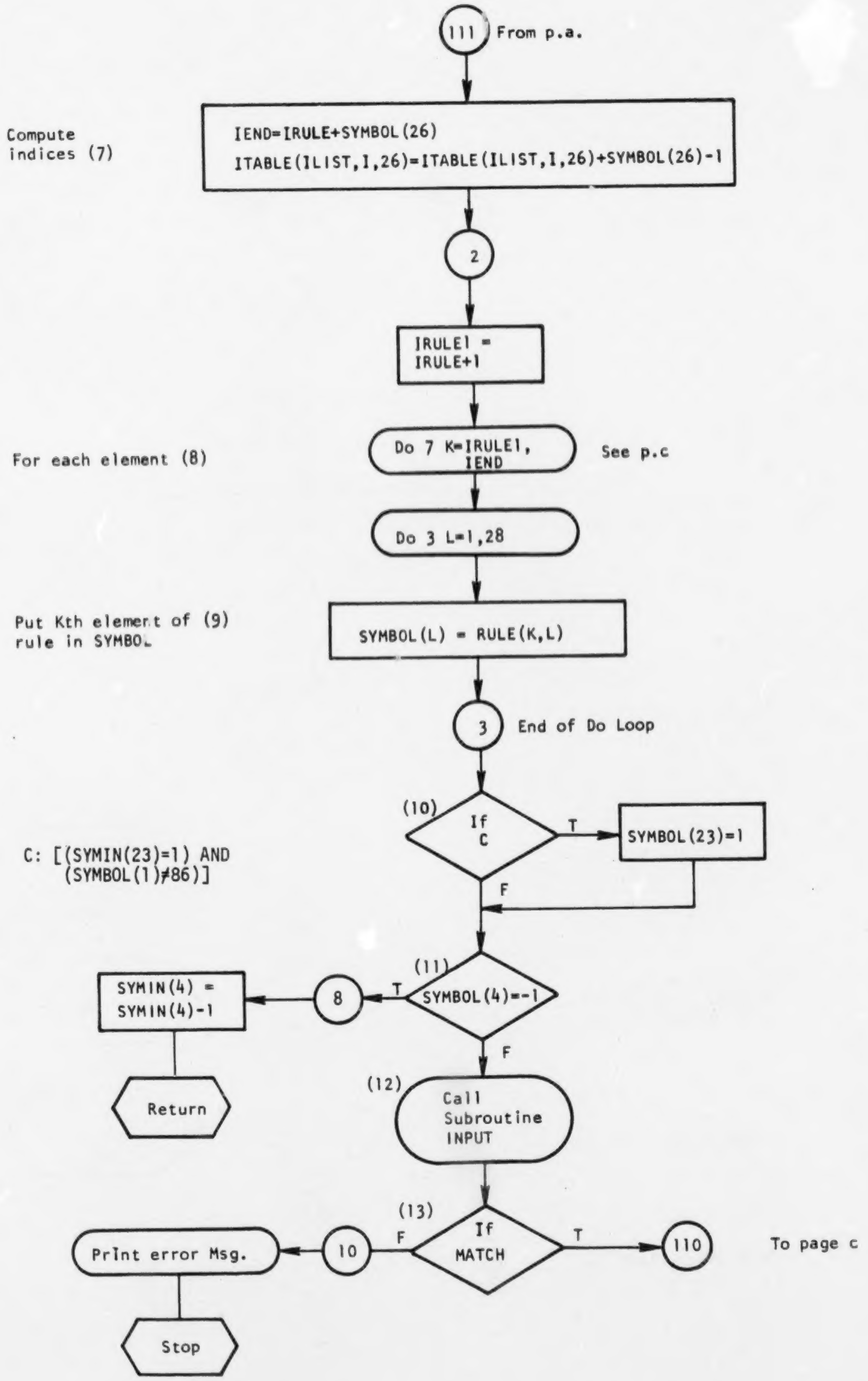

111
From p.a.
Compute indices (7)
IEND=IRULE+SYMBOL(26)
ITABLE(ILIST,I,26)=ITABLE(ILIST,I,26)+SYMBOL(26)-1
2
IRULE1 =
IRULE+1
For each element (8)
Do 7 K=IRULE1, IEND
See p.c
Do 3 L=1,28
Put Kth element of (9) rule in SYMBOL
SYMBOL(L) = RULE(K,L)
3
End of Do Loop
(10)
If C
T
SYMBOL(23)=1
F
C: [(SYMIN(23)=1) AND (SYMBOL(1)≠86)]
(11)
SYMBOL(4)=-1
T
8
SYMIN(4) = SYMIN(4)-1
Return
F
(12)
Call Subroutine INPUT
(13)
If MATCH
T
110
To page c
F
10
Print error Msg.
Stop

Figure 3.14b Subprogram RULRIT

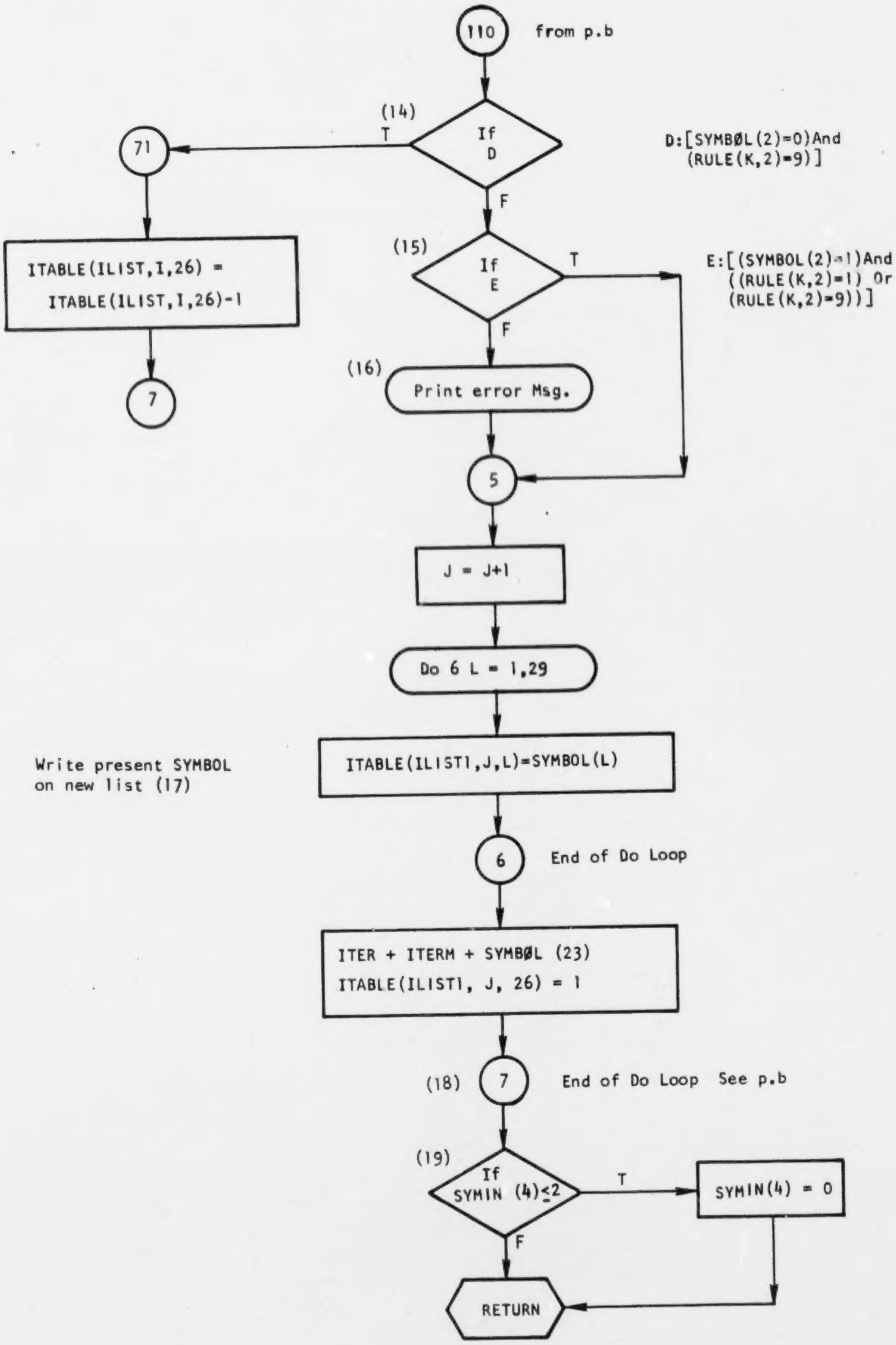

110
from p.b
(14)
If
D
T
F
D:[SYMBØL(2)=0)And
(RULE(K,2)=9)]
71
ITABLE(ILIST,I,26) =
ITABLE(ILIST,I,26)-1
7
(15)
If
E
T
F
E:[(SYMBOL(2)=1)And
((RULE(K,2)=1) Or
(RULE(K,2)=9))]
(16)
Print error Msg.
5
J = J+1
Do 6 L = 1,29
Write present SYMBOL
on new list (17)
ITABLE(ILIST1,J,L)=SYMBOL(L)
6
End of Do Loop
ITER + ITERM + SYMBØL (23)
ITABLE(ILIST1, J, 26) = 1
(18)
7
End of Do Loop See p.b
(19)
If
SYMIN (4)≤2
T
F
SYMIN(4) = 0
RETURN

Figure 3.14c Subprogram RULRIT

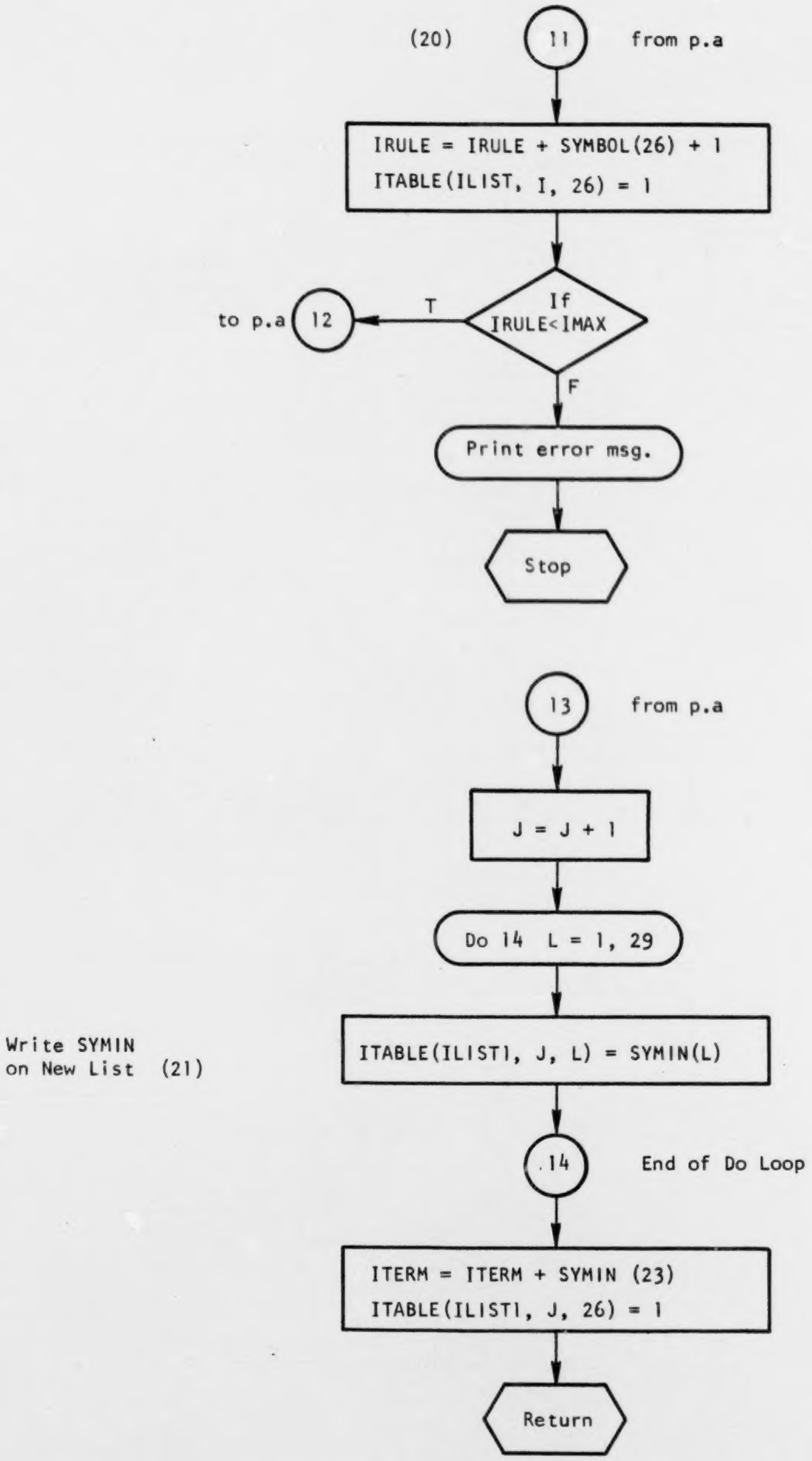

(20)
11
from p.a
IRULE = IRULE + SYMBOL(26) + 1
ITABLE(ILIST, I, 26) = 1
If
IRULE<IMAX
T
to p.a
12
F
Print error msg.
Stop
13
from p.a
J = J + 1
Do 14 L = 1, 29
Write SYMIN
on New List (21)
ITABLE(ILIST1, J, L) = SYMIN(L)
14
End of Do Loop
ITERM = ITERM + SYMIN (23)
ITABLE(ILIST1, J, 26) = 1
Return

Figure 3.14d Subprogram RULRIT

(21) Write the "present constituent," which is a simple, non-negated terminal symbol, on the new list and return to the calling program.

3.3.1.13 Subprogram SYMACH

This subprogram (Figure 3.15) is called by Subprogram RULRIT (step 3) to check that the "present rule symbol" matches the "present constituent" with respect to the criteria of the grammar. The subprogram performs the following functions:

(1) Initializes the logic monitor MATCH to "true."

(2) If the symbol types do not match or if the "present rule symbol" is a variable symbol, skip to step 11.

(3) For each subscript up to 22 perform steps 4 through 10.

(4) If subscript 3, and the "present rule symbol" is a variable symbol, skip to step 10.

(5) If subscript 7, and the "present rule symbol" is not a negative, skip to step 10.

(6) If the value of the given subscript of the "present rule symbol" is ≧ 0 and ≦ 9, that is, it is a non-dependent variable, skip to step 10.

(7) If the value of the given subscript of the "present rule symbol" equals the value of the subscript of the "present constituent," skip to step 10.

(8) If the "present constituent" is not negated or compounded, skip to step 11.

(9) If the given subscript is an independent variable for the "present rule symbol" and the value of the given subscript is zero for the "present constituent", continue to step 10, otherwise skip to step 11.

(10) If not the last subscript, advance the subscript index and return to step 3, otherwise return to the calling program.

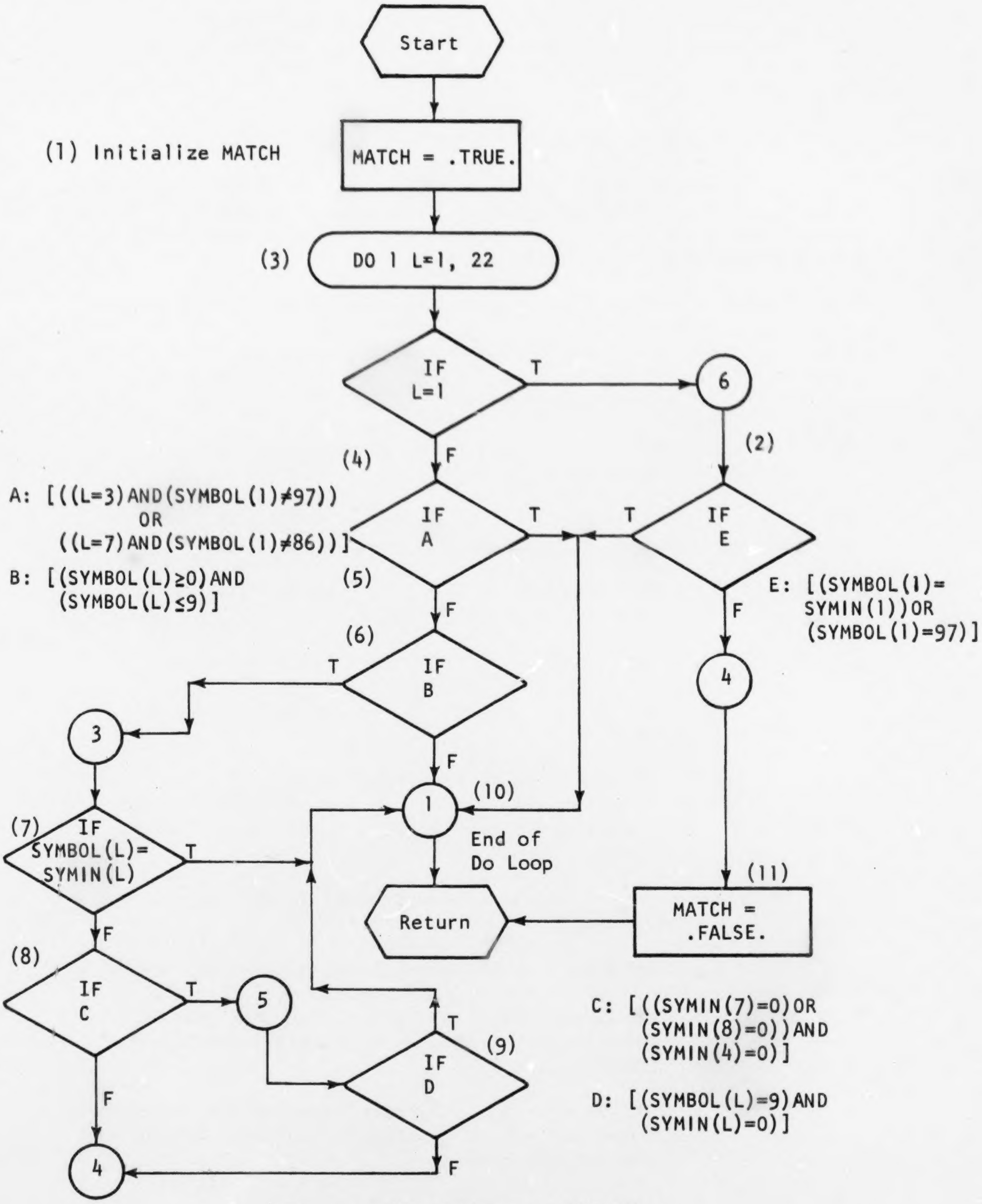

Start
(1) Initialize MATCH
MATCH = .TRUE.
(3)
DO 1 L=1, 22
IF L=1
T
6
(2)
F
(4)
A: [((L=3) AND (SYMBOL(1)≠97)) OR ((L=7) AND (SYMBOL(1)≠86))]
IF A
T
T
IF E
B: [(SYMBOL(L)≥0) AND (SYMBOL(L)≤9)]
(5)
E: [(SYMBOL(1)= SYMIN(1)) OR (SYMBOL(1)=97)]
F
F
(6)
IF B
T
4
3
F
(10)
1
(7)
IF SYMBOL(L)= SYMIN(L)
T
End of Do Loop
(11)
Return
MATCH = .FALSE.
F
(8)
IF C
T
5
T
C: [((SYMIN(7)=0) OR (SYMIN(8)=0)) AND (SYMIN(4)=0)]
(9)
IF D
F
D: [(SYMBOL(L)=9) AND (SYMIN(L)=0)]
4
F

Figure 3.15 Subprogram SYMACH

(11) Set logic monitor MATCH to "false" and return to the calling program.

3.3.1.14 Subprogram VARATT

This subprogram (Figure 3.16) is called by Subprogram RULRIT (step 4) to compute the value of the various dependent variable subscripts for the "present rule symbol." For a given rule, certain of the subscripts may be designated as dependent variables--that is, the values of these subscripts depend on the values of the corresponding subscripts in the "present constituent." These dependent variable subscripts are written as lower case alphabetic characters in the rules; these were transformed to integer numbers when the rules were read in by the main program (step 3). Reference in Table 3-1 indicates that all alphabetic characters have been transformed to an integer value greater than 9, so that this test is used to identify dependent variable subscripts.

This subprogram performs the following operations:

(1) Initializes the dependent variable storage register VARSTR to erase previous data.

(2) If the "present rule symbol" is a variable symbol (used in rules for negating or compounding), set the symbol type subscript of the "present rule symbol" to the same value as that of the "present constituent" and store this value in VARSTR.

(3) For subscripts 2 through 22, perform steps 4 through 6.

(4) If the given L-th subscript of the "present rule symbol" is an independent variable ($0 \leqq \text{SYMBOL(L)} \leqq 9$) skip to step 6.

(5) If the value of the given subscript of the "present rule symbol" is a negative number, convert it to a positive number, otherwise set it to the value of the corresponding subscript of the "present constituent," and in either case store the new value of this subscript in VARSTR.

(6) If not the 22-nd subscript, advance index L and return to step 4, otherwise continue.

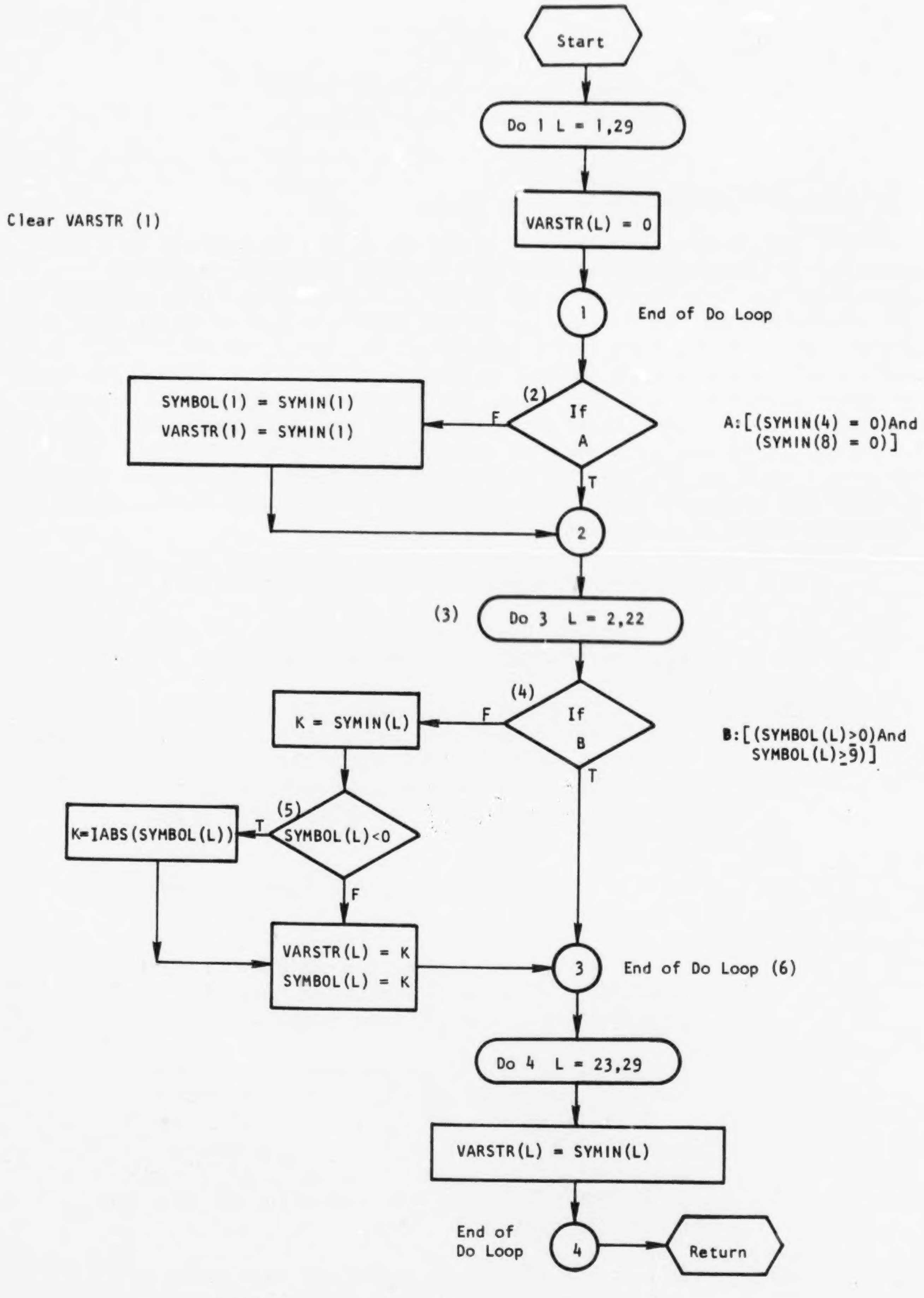

Start
Do 1 L = 1,29
Clear VARSTR (1)
VARSTR(L) = 0
1
End of Do Loop
(2)
If
A
F
T
SYMBOL(1) = SYMIN(1)
VARSTR(1) = SYMIN(1)
A:[(SYMIN(4) = 0)And
(SYMIN(8) = 0)]
2
(3)
Do 3 L = 2,22
(4)
If
B
F
T
K = SYMIN(L)
B:[(SYMBOL(L)≥0)And
SYMBOL(L)≥9)]
(5)
SYMBOL(L)<0
T
F
K=IABS(SYMBOL(L))
VARSTR(L) = K
SYMBOL(L) = K
3
End of Do Loop (6)
Do 4 L = 23,29
VARSTR(L) = SYMIN(L)
End of
Do Loop
4
Return

Figure 3.16 Subprogram VARATT

(7) For subscripts 23 through 29, store the values of the corresponding subscripts of the "present constituent" in VARSTR and return to the calling program.

3.3.1.15 Subprogram INFLEX

This subprogram (Figure 3.17) is called by Subprogram ORTHOG (step 4) to compute the correct spelling of a given terminal symbol (word) of the grammar. The subprogram is based on the following model of modern Hebrew words:

$$W = LBS_1S_2S_3S_4S_5S_6S_7S_8S_9S_{10}S_{11}S_{12}A_1A_2A_3A_4R$$

where W is a Hebrew word with up to 19 characters. The right hand symbols represent the following word constituents:

L = left hand word divider

B = inflection prefix

S_1 = stem prefix 1

S_2 = stem prefix 2

S_3 = root letter 1

S_4 = stem infix 1

S_5 = stem infix 2

S_6 = root letter 2

S_7 = stem infix 3

S_8 = root letter 3

S_9 = stem infix 4

S_{10} = root letter 4

S_{11} = stem suffix 1

S_{12} = stem suffix 2

A_1 = inflection suffix 1

A_2 = inflection suffix 2

A_3 = inflection suffix 3

A_4 = inflection suffix 4

R = right hand word divider

Each of these constituents is dependent on one or more of the following subscripts for its orthographic value:

a - inflection class

b - stem class

n - number

g - gender

p - person

r - preposition class

m - mood

t - tense

j - state

w_1 - root letter 1

w_2 - root letter 2

w_3 - root letter 3

w_4 - root letter 4

h - fem. sing. class

x - number-gender transformation class

f - word divider class

The subprogram computes the orthographic value of each word constituent as a function of the input values of the subscripts and assembles the resultant word in a left-justified format in array WORD.[12]

The subprogram has the option of reading in the full spelling of words that do not follow the inflectional laws of Hebrew. It also computes the orthography of the following types of words without the use of external dictionary data:

(a) demonstrative pronouns

(b) personal pronouns (nominative)

(c) personal pronouns (accusative)

(d) the relative pronoun

The flow chart (Figure 3.17) is provided to outline the major steps of the subprogram. The chart is self explained.

3.3.2 Input Map for Computer Program

This section describes the input data required to use the computer program of the Algorithm for Generating Hebrew Sentences. Three types of data cards are required:

(1) Program option card,

(2) Data cards describing the grammar and its rules.

(3) Data cards describing the sentences to be generated.

The following sections describe each type of card.

---

[12] The logic of the computations is not included here, but it is explained in J. Price's "An Algorithm for Generating Hebrew Words," Computer Studies in the Humanities and Verbal Behavior, Vol. 1, No. 2 (1968).

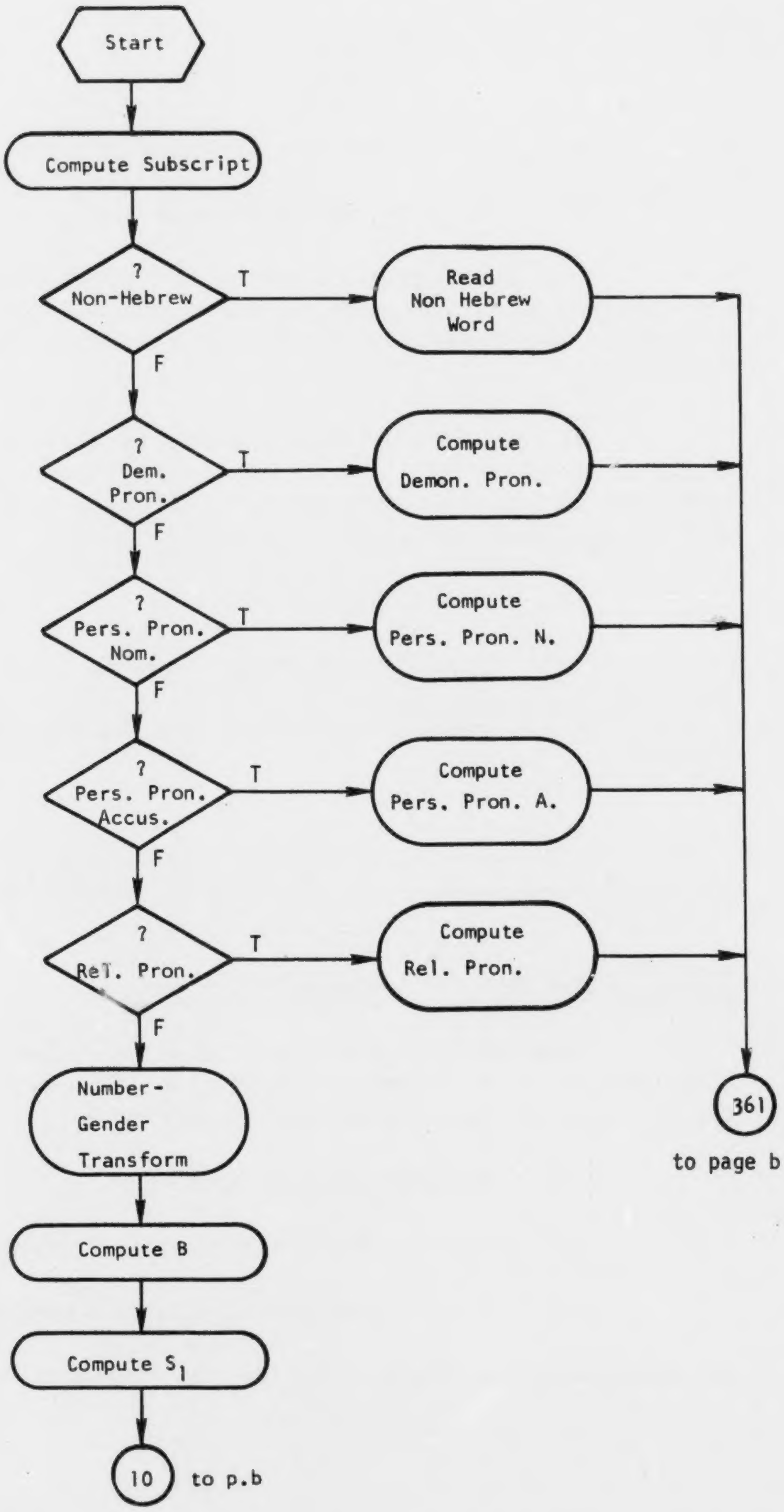

Start
Compute Subscript
? Non-Hebrew
T
F
Read Non Hebrew Word
? Dem. Pron.
Compute Demon. Pron.
? Pers. Pron. Nom.
Compute Pers. Pron. N.
? Pers. Pron. Accus.
Compute Pers. Pron. A.
? Rel. Pron.
Compute Rel. Pron.
Number-Gender Transform
Compute B
Compute S1
10
to p.b
361
to page b

Figure 3.17a Subprogram INFLEX

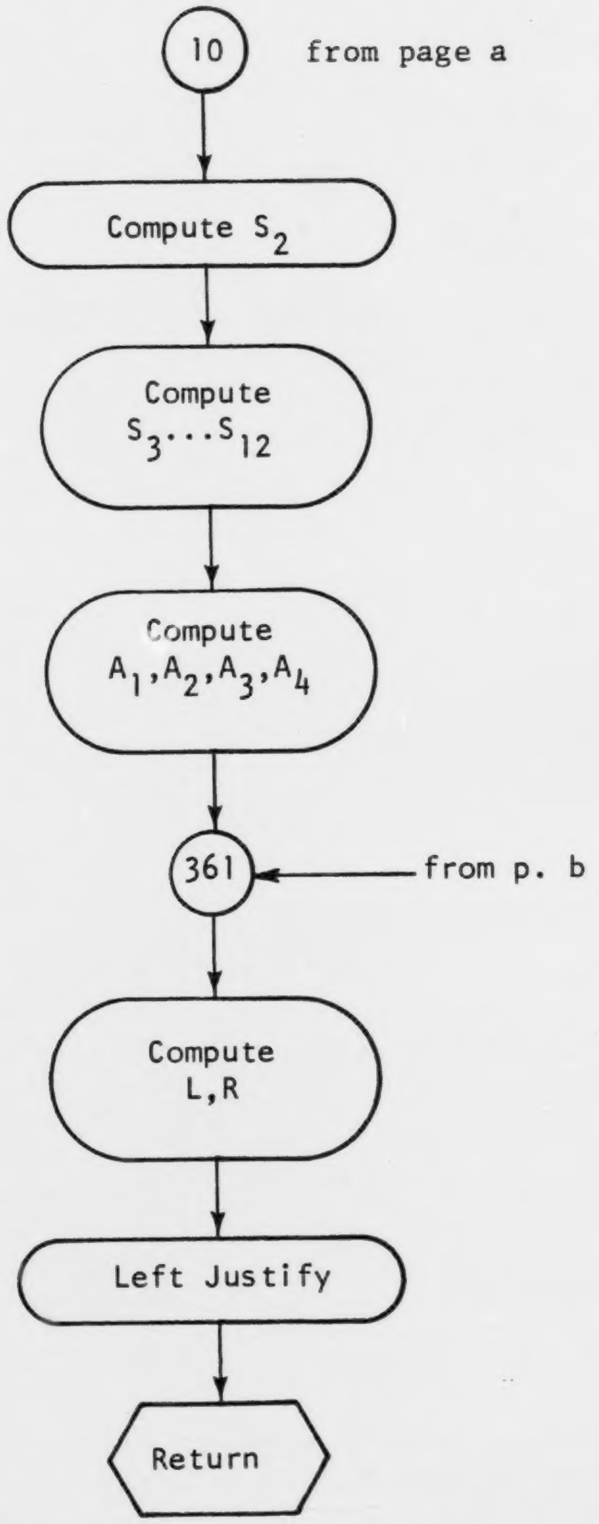

10
from page a
Compute S_2
Compute $S_3 \ldots S_{12}$
Compute A_1, A_2, A_3, A_4
361
from p. b
Compute L,R
Left Justify
Return

Figure 3.17b Subprogram INFLEX

3.3.2.1 Program Option Card

This card is used to control the various options available in the program. The data and the associated options are as follows:

Card Col.	Format	Variable	Description
1-5	I5	MODE	= 0, grammar rules are to be read in from cards
			= 1, grammar rules are stored on tap ITAPE.
6-10	I5	ITAPE	logical unit number of magnetic tape for storage of grammar rules.
11-15	I5	ITRACE	= 0, no diagnostic messages to be printed out.
			= 1, print out diagnostic messages of program.
16-20	I5	IOUTPT	= 0, do not print out full description of nodes at every level of synthesis.
			1, print out full description of nodes at every level of synthesis.
21-25	I5	IOUTRE	= 0, do not print out tree diagram of snythesis,
			= 1, print out tree diagram.

The variable MODE is used to control the source of the grammar rules used by the program. Initially the rules are read in from cards (MODE=0) and stored on the magnetic tape mounted on logical unit ITAPE. On subsequent runs the rules are read from the tape (MODE=1) unless changes in the rules are to be made.

Variable ITAPE defines the logical unit number containing the magnetic tape on which the grammar rules are stored (MODE=1) or are to be stored (MODE=0).

Variable ITRACE is used for tracing the progress of the program throughout its operation. If ITRACE=1, the program prints out numerous messages together with the values of certain critical variables at key steps of the program. The use of this variable is for program debugging only, it should normally be set to zero.

Variable IOUTPT is used to control the output of the computer. The computer will list a complete description of each nodal point at every level of the synthesis if IOUTPT = 1 (see Appendix C for a sample of this output). This output is useful when the user wants to examine the output results in fine detail, otherwise IOUTPT should be zero.

Variable IOUTRE is used to control the output of the computer. The computer will construct and print out a tree diagram of the synthesized sentence. Each nodal point is identified as to symbol type and class only (see Section 2.3 or Part II of this report for a sample of this output). If more detailed data are required the IOUPUT option should be used also. Note: either IOUTPT or IOUTRE (or both) must be 1.

3.3.2.2 Program Grammar Cards

The program grammar cards consists of four types:

Type 2-1: Transliteration Table

Type 2-2. Grammar Rules

Type 2-3: Restraint Matrix

Type 2-4: The Table of Symbols

The program grammar cards are not required if MODE=1; in this case these data are read from the magnetic tape (ITAPE). The following sections define the program grammar cards.

3.3.2.2.1 Transliteration Table: Card 2-1

This card defines the transliteration Table (see Table 3-1). The data are as follows:

Card Col.	Format	Variable	Description
1-6	I6	NL	= number of elements in Table (50 max.)
7-56	50A1	TRANSL(L) L=1,NL	Transliteration Table

3.3.2.2.2 Grammar Rules: Cards 2-2

These cards define the rules of the grammar. Each card defines one constituent of a rule. Thus the rule

$$A = B + C$$

requires three cards: the first defines A, the second B, and the third C. A maximum of 800 cards is permitted. A card with 999 in columns 1-3 should follow the last grammar rule card. The rules should be placed in numeric order by rule number. See Appendix A for a complete listing of the rules used for the present research. The following is the data format for each card:

Card Col.	Format	Variable	Description
1-3	A3	SN	Symbol number (see Table 3-2)
6	A1	M	Subscript m - optional/mandatory
9	A1	F	Subscript f - compound class
11-12	A2	K	Subscript k - compound number
15	A1	B	Subscript b - compound type
18	A1	C	Subscript c - symbol class
21	A1	ℓ	Subscript ℓ - negative class
24	A1	Y	Subscript y - negative/positive
27	A1	D	Subscript d - definiteness
30	A1	N	Subscript n - number
33	A1	G	Subscript g - gender
36	A1	P	Subscript p - person
39	A1	R	Subscript r - preposition class
42	A1	A	Subscript a - verb class
45	A1	V	Subscript v - voice
48	A1	I	Subscript i - mood
51	A1	T	Subscript t - tense
53-54	A2	S	Subscript s - stem
57	A1	W1	Subscript w_1 - root letter 1
60	A1	W2	Subscript w_2 - root letter 2
63	A1	W3	Subscript w_3 - root letter 3

Card Col.	Format	Variable	Description
66	A1	W4	Subscript w_4 - root letter 4
69	A1	ST	Symbol Type
70-72	I3	RN	Rule Number
73	I1	EN	Element Number
74	I1	NRH	No. of Rt. Hand Elements
75-76	I2	RT	Restraint Type
77-78	I2	RS	Restraint Subscript

Variable SN is the symbol number of the given rule element as specified on Table 3-2. A card with SN=999 should follow the last grammar rule.

Variables m, f, k, b, c, ℓ, y, d, n, g, p, r, a, v, i, t, s, w_1, w_2, w_3, and w_4 are the subscripts of the symbols as defined in Part II, Section 2.2.1.

Variable ST identifies the symbol as follows:

ST = 0 for non-terminal symbols

ST = 1 for terminal symbols

Variable RN is the rule number associated with the given card. For example: in the rule

$$A = B + C \qquad (52.1)$$

Variable RN = 521 for all cards of the rule. Variable EN is the element number of the symbol defined by the card. In the above example:

on the card defining A, EN=0

on the card defining B, EN=1

on the card defining C, EN=2

Variable NRH defines the number of right hand elements in the rule. In the above example, NRH=2 for all cards of the rule.

Variable RT defines the restraint type that applies to the symbol. The value of RT refers to a row of restraints in the Restraint

Matrix which is defined by Cards 2-3. Variable RS defines the subscript of the symbol to which the restraint applies. For example,

RT = 3, RS = 10, means that restraint type 3 applies to the 10-th subscript n.

3.3.2.2.3 Restraint Matrix: Cards 2-3

These cards define the restraints that are placed on a given symbol in a grammar rule. See Section 3.3.1.7 for a discussion and explanation of the data in Matrix RESTRT. The following is the format of the data:

Card Col.	Format	Variable	Description
1-5	I5	IM	Number of restraints on I-th list (Max. = 4)
6-25	4I5	RESTRT(J,I) J=2, IM + 1	I-th list of restraint codes

Number of cards 15.

The data on these cards are listed in Table 3-5

Table 3-5

RESTRAINT MATRIX DATA

CARD COL.					CARD NO.
1-5	6-10	11-15	16-20	21-25	
1	-1	0	0	0	2-3.1
1	-2	0	0	0	2-3.2
1	-3	0	0	0	2-3.3
2	0	1	0	0	2-3.4
2	1	2	0	0	2-3.5
2	-1	-2	0	0	2-3.6
2	1	3	0	0	2-3.7
2	2	5	0	0	2-3.8
2	4	5	0	0	2-3.9
2	6	7	0	0	2-3.10
3	1	2	3	0	2-3.11
3	4	5	6	0	2-3.12
4	1	2	3	4	2-3.13
4	-1	-2	-3	-4	2-3.14
0	0	0	0	0	2-3.15

3.3.2.2.4 The Table of Symbols: Cards 2-4

These cards define the symbols used in the grammar. See Section 3.1.1.2 for further discussion. Ten cards are required for these data which are in 10(2x,A3) format. The data for the Hebrew grammar are given in Table 3-6.

Table 3-6

TABLE OF SYMBOLS

Card Column										Card No.
0-5	6-10	11-15	16-20	21-25	26-30	31-35	36-40	41-45	46-50	
Z-	VQO	APA	AP	AS	NA	SQO	NS	RD	RO	2-4.1
BAA	BAB	BAC	BAD	BAE	BAF	BBA	BBB	BA	BBC	2-4.2
BP	NPB	NPA	NPC	NAP	NP	DPA	DPB	DPC	DP	2-4.3
EA	VB	VBB	VC	VAA	VA	BC	XP	NO	DPD	2-4.4
RSP	NSP	NOP	NIP	NPX	VMA	VMB	VMC	VMD	VM	2-4.5
VMR	VMI	VP	VRB	VRI	NV	NW	EPB	EPA	EP	2-4.6
SAA	SAB	SAC	SA	SRO	SRI	RG	KN	KC	KK	2-4.7
KI	KD	SD	S-	SI	SC	A-	B-	C-	D-	2-4.8
E-	G-	H-	I-	J-	L-	N-	O-	P-	Q-	2-4.9
R-	T-	U-	V-	W-	Y-	F-	98	99	100	2-4.10

3.3.2.3 Sentence Description Data Cards

The sentence generating algorithm uses a set of variables the values of which fully define sentences to be generated. The input data for a given sentence constitute the set of values for the variables that produce that sentence.

The input to the algorithm is equivalent to the data supplied by a Transfer Grammar in an automatic system. These data provide a complete grammatical analysis of the sentence transformed into its Hebrew equivalent and presented in the order required by the Hebrew grammar. The input also represents the approximate output of an algorithm for analyzing Hebrew sentences which would be the input to a Transfer Grammar from Hebrew to some target language; it is the equivalent of the "information" supplied by the information system originating the message.

This section provides a guide for preparing sentence description input data. It defines the order of the input data for the sentence generating algorithm. It arranges the data in the order required by the rules of the algorithm.

The guide consists of a list of the types of data cards and a set of instructions for using the list to produce a set of input data cards for a given sentence to be generated. The list contains the following information for each type of data card:

(1) card type number

(2) name of the grammatical function of the associated data

(3) list of the variables to be assigned a value, and the associated meaning of each value

(4) list of new card types to be included in the data deck

(5) format for data

The complete list of sentence description data card types is contained in Appendix D.

The instructions for using the list are given in flow chart form in Figure 3.18. The following sections explain the operations of each block of the figure in detail.

(1) Block 1 - The input data always begins with card type 76 for each sentence to be generated. This card specifies the type of sentence to be generated. The node number is 1. Go to Block 2.

(2) Block 2 - For a given card type the input list specifies the data options that are available for the sentence element being defined. The numbers associated with selected options are coded in the card columns designated by the input map. The given card type is assigned a node number that is one greater than the preceding card. Go to Block 3.

(3) Block 3 - If the card presently being coded indicates that the associated sentence element is compounded (i.e. K>1), the instruction path goes to Block 4, otherwise to Block 5.

(4) Block 4 - When the card being coded indicates compounding (i.e., K>1), the present card type is written at the end of the list K times and the governing node number of these K cards is made as the node number of the card presently being coded. Go to Block 7.

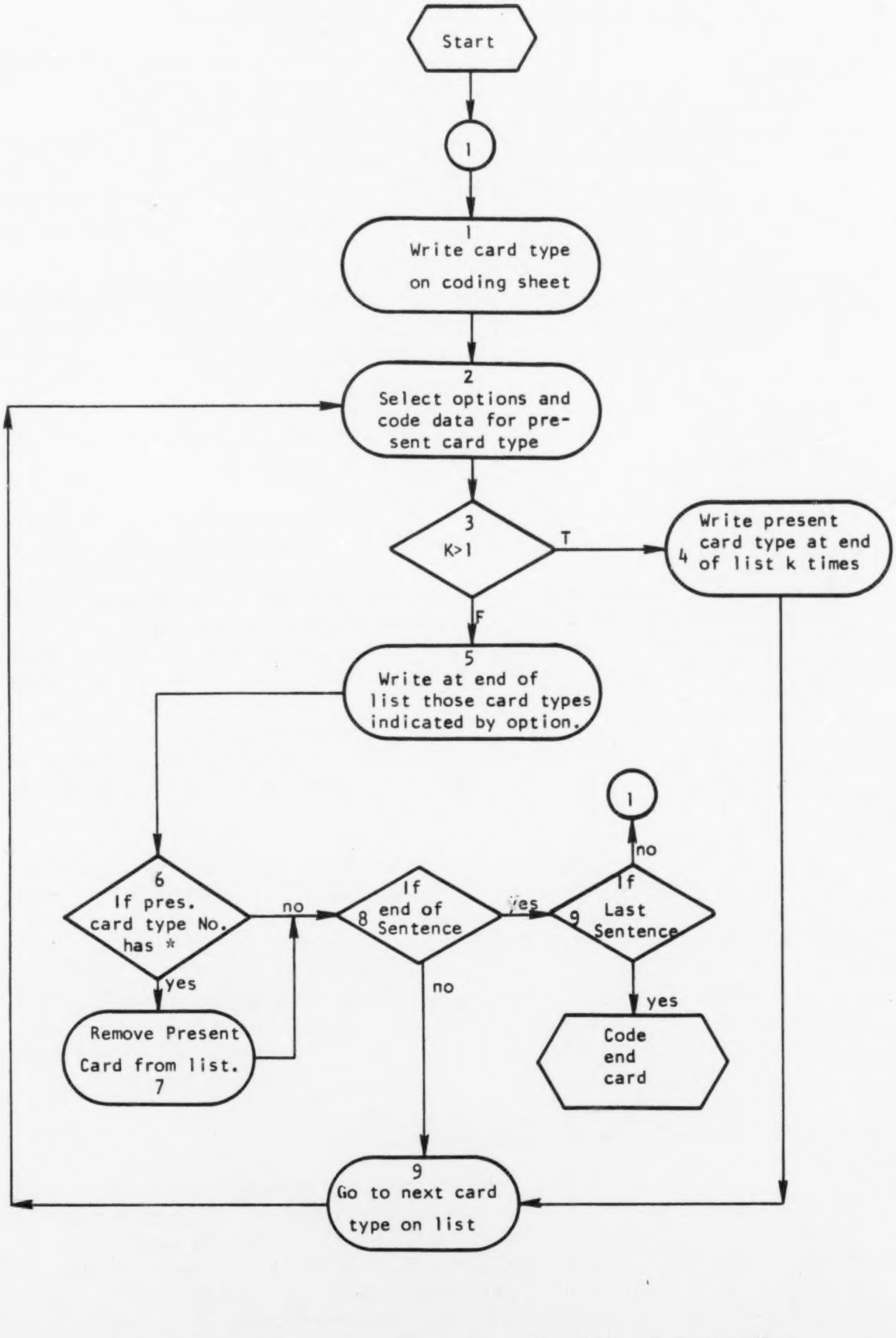

Start
1
1
Write card type on coding sheet
2
Select options and code data for present card type
3
K>1
T
F
4
Write present card type at end of list k times
5
Write at end of list those card types indicated by option.
6
If pres. card type No. has *
yes
no
Remove Present Card from list.
7
8
If end of Sentence
yes
no
9
If Last Sentence
no
1
yes
Code end card
9
Go to next card type on list

Figure 3.18 Flow Chart of Instructions for Use of Input Map

(5) Block 5 - For some options available for the card type presently being coded, the input map specifies new card types to be added to the end of the list of input data cards. These card types should be written at the end of the list in the specified sequence. Their governing node number is that of the card presently being coded. Any special limitations for these added card types should be immediately coded as specified by the input map. Go to Block 6.

(6) Block 6 - Certain card type numbers are written with an asterisk (example: Card Type: 8*). These cards do not provide new input data but are required in order to add new card types to the list in the proper order. Once these cards have been coded and the new card types added to the end of the list, they should be removed from the list. If the card type number has an asterisk, the instruction path goes to Block 7, otherwise to Block 8.

(7) Block 7 - If the present card type number has an asterisk (*) remove the card from the list and procede to Block 8.

(8) Block 8 - The end of the sentence is reached when the card presently being coded is the last card on the list and the input list does not specify any further card types. If the end of the sentence is reached, the instruction path goes to Block 10, otherwise to Block 9.

(9) Block 9 - If the coding is completed for the present card type and it is not the last card type on the list, move to the next card type on the list and go to Block 2.

(10) Block 10 - The input list may specify more than one sentence in a given exercise. If the end of a sentence has been reached and there are more sentences to be coded in the exercise go to Block 1, otherwise Block 11.

(11) Block 11 - When coding is completed for all sentences in the exercise, the last card on the list should be coded with card type number 999. This terminates the input data.

3.4 Test and Demonstration of Algorithm

In order to test the computerized algorithm and to verify the grammar of modern Hebrew syntax, a total of 47 sentences were generated on the computer. These sentences consist of various permutations to two arbitrarily composed sentences and of 14 sentences selected from Elias Persky and Sol Scharfstein, *SHOLOM YISROEL, Part Two*, Ktav Publishing House, Inc., 1962.

Originally it was planned to select 25 different sentences, but it was found that more grammar rules and symbols could be tested and demonstrated by using permutations. In a few cases, the permutations are grammatical but do not make good sense; these were selected purely for the purpose of testing some specific rule. Some permutations illustrate classical Hebrew options of the grammar. Of the 47 sentences generated, 42 are grammatically correct, three have errors requiring modification of the grammar rules (7g, 10a, and 11c), and two are correct except for a superfluous period (26a, 26aa).

Early computations were performed on a UNIVAC 1107; which was replaced by a UNIVAC 1108 for more recent work. No record was kept of the computing time required to generate a sentence, however, a conservative estimate would be about 2.0 seconds of computer time per sentence. Computing time is roughly proportional to the length of the sentence.

3.4.1 Sentences Generated By Algorithm

This section contains a list of the Hebrew sentences generated by the computerized algorithm preceded by their English equivalent. The number accompanying each pair of sentences refers to the corresponding tree diagrams generated by the algorithm which are contained in Section 2.3.1 of Part II of this report.

(1) Chaim Nachman Biyalik was a great poet in the land of Israel. XYYM NXMN BYALYQ HYH MSWRR GDWL BAR& YSRAL.

(1a) Was Chaim Nachman Biyalik a great poet in the land of Israel? HAM XYYM NXMN BYALYQ HYH MSWRR GDWL BAR& YSRAL?

(1b) Chaim Nachman Biyalik be a great poet in the land of Israel! XYYM NXMN BYALYQ HYH[12] MSWRR GDWL BAR& YSRAL!

(1c) Chaim Nachman Biyalik will be a great poet in the land of Israel. XYYM NXMN BYALYQ YHYH MSWRR GDWL BAR& YSRAL.

(1dd) Chaim Nachman Biyalik is a great poet in the land of Israel. XYYM NXMN BYALYQ MSWRR GDWL BAR& YSRAL.

---

[12] This illustrates the imperative sentence. The HYH is the imperative הֱיֵה! .

(1ddd) Chaim Nachman Biyalik is a great poet in the
land of Israel. XYYM NXMN BYALYQ HWA MSWRR
GDWL BAR& YSRAL.

(1e) Chaim Nachman Biyalik is not a great poet in
the land of Israel. AYN XYYM NXMN BYALYQ
MSWRR GDWL BAR& YSRAL.

(1ee) Chaim Nachman Biyalik is not a great
poet in the land of Israel. XYYM NXMN
BYALYQ AYNW MSWRR GDWL BAR& YSRAL.

(2) He was greater than all the poets.
HWA HYH GDWL MKL HMSWRRYM.

(2a) Who was greater than all the poets?
MY HYH GDWL MKL HMSWRRYM?

(2b) He is greater than all the poets.
YSNW GDWL MKL HMSWRRYM.

(4) He lived in a small village.
HWA YSB BKPR Q@N.

(4a) He will live in a small village.
HWA YYSB BKPR Q@N.

(4b) He is living in a small village.
HWA YWSB BKPR Q@N.

(4c) He used to sit on a small chair.
HWA HYH YWSB OL KCA Q@N.

(4d) He will continually live in a small villiage.
HWA YHYH YWSB BKPR Q@N.

(5) He greatly loved to study law.
HWA AHB MAWD LLMWD TWRH.

(6) But he also loved the fields and the forests.
ABL HWA AHB GM AT HSDWT WAT HYORYM.

(6a) But what did he love?
ABL MH HWA AHB?

(7) When Biyalik grew up, he went to study in a large
academy. KASR BYALYQ GDL, HLK LLMWD BYSYBH GDWLH.

(7a) When Biyalik grew up, did he go to study in a large academy? KASR BYALYQ GDL, HHLK[13] LLMWD BYSYBH GDWLH?

(7aa) When Biyalik grew up, did he go to study in a large academy? KASR BYALYQ GDL, HAM[14] HLK LLMWD BYSYBH GDWLH?

(7b) Did Biyalik go to study in a large academy when he grew up? HAM BYALYQ HLK LLMWD BYSYBH GDWLH, KASR GDL?

(7c) If Biyalik had grown up, he would have gone to study in a large academy. LW BYALYQ GDL, HLK LLMWD BYSYBH GDWLH.

(7d) If Biyalik had not grown up, he would not have gone to study in a large academy. LWLA BYALYQ GDL, LA HLK LLMWD BYSYBH GDWLH.

(7e) If Biyalik will not grow up, he will not go to study in a large academy. AM BYALYQ LA YGDL, LA YYLK LLMWD BYSYBH GDWLH.

(7f) If Biyalik will grow up, he will to to study in a large acedemy. AM BYALYQ YGDL, YYLK LLMWD BYSYBH GDWLH.

(7g) In order that Biyalik should grow up, he went to study in a large academy. LMON[15] BYALYQ YGDL, HLK LLMWD BYSYBH GDWLH.

(8) In this academy he wrote the first poems. BYSYBH HZWT HWA KTB AT HSYRYM HRASWNYM.

(9) In these poems he wrote about the sun and the moon. BSYRYM HALH HWA KTB OL HSMS WOL HYRX.

(10) He wrote about the fields and forests which he saw in the village. HWA KTB OL HSDWT WOL HYORYM ASR[16] RAH AWTM[17] BKPR.

---

[13] This illustrates the classical Hebrew option.

[14] This illustrates the modern Hebrew option.

[15] Error, This should be KDY SBYALQ...

[16] This is classical Hebrew. Modern Hebrew prefers SRAH AWTM, or SAWTM RAH.

[17] The use of AWTM here is classical and not necessary in modern Hebrew.

(10a) He wrote about the fields and forests which he saw in the village. HWA KTB OL HSDWT WOL HYORYM ASR[18] HWA RAH AWTM[17] BKPR.

(11) He wrote many poems about the land of Israel and the people of Israel. HWA KTB HRBH SYRYM OL AR& YSRAL WOL OM YSRAL.

(11a) He loved the land of Israel about which he wrote many poems. HWA AHB AT AR& YSRAL ASR[19] KTB OLYH HRBH SYRYM.

(11c) About what did he write many poems?
OL MH HWA KTB OLYHM[20] HRBH SYRYM?

(12) The students studied these poems and knew them by heart. HTLMYDYM LMDW AT HSYRYM HALH WYDOW AWTM OL PH.

(12a) The students who studied the poems knew them by heart. HTLMYDYM ASR[21] LMDW AT HSYRYM HALH YDOW AWTM OL PH.

(13) In the year 1924 Biyalik traveled to the land of Israel. BSNT 1924 BYALYQ NCO LAR& YSRAL.

(23) All the Jews wept over his death, because there was no poet like him in Israel. KL HYHWDYM BKW OL MWTW, KY[22] LWA HYH MSWRR KMWHW BYSRAL.

(26a) One time Grandfather asked Uri, "do you want to see the synagogue?" HCB SAL AT AWRY POM: "HAM ATH RW&H LRAWT AT BYT HKNCT?".[23]

(26aa) One time Grandfather asked Uri, "do you want to see the synagogue?" HCB SAL AT AWRY POM: "HATH[24] RW&H LRAWT AT BYT HKNCT?".[23]

---

[18] Error. It should be SHWA RAH...

[19] This is classical Hebrew. Modern Hebrew prefers SOLYH KTB.

[20] Error. There should be no OLYHM. This requires a modification of the grammar rules.

[21] This is classical Hebrew. Modern Hebrew prefers SLMDW.

[22] This is classical Hebrew. Modern Hebrew prefers MPNY SLA HYH...

[23] There should be no period.

[24] This illustrates the classical Hebrew option, but HRW&H ATH...is preferred.

(26b) One time Grandfather said that he wanted to see the synagogue. HCB AMR POM AXD[25]KY[26] HWA RW&H LRAWT AT BYT HKNCT.

(26c) One time Grandfather said that he wanted to see the synagogue. HCB AMR POM SHW RW&H LRAWT AT BYT HKNCT.

(A) The children, Uri and Rebeccah, are sitting on a chair and reading their books. HYLDYM, AWRY WRBKH, YWSBYM OL KYCA WQWRAYM AT HCPRYM SLHM.

(101a) Joseph does not have a room.
AYN LYWCP XDR.

(101b) Joseph has a room.
YS LYWCP XDR.

(101c) Joseph will not have a room.
LYWCP LWA YHYH XDR.

3.4.2 Grammar Rules Tested By The Alorithm

The 47 sentences generated by the computerized algorithm were selected to test the algorithm and to verify the rules of the grammar of modern Hebrew syntax. In generating these sentences, 113 of the 179 rules on the intermediate symbols were tested and demonstrated, together with 43 of the 65 class variations of the terminal symbols. Of the 73 intermediate symbols, 61 were tested for a least one class, and 34 were tested for all class variants. Of the 20 terminal symbols, 17 were tested for at least one class, and 12 were tested for all class variants.

Table 3-7 is a listing of the sentence numbers of the test sentences that demonstrate the symbols of the grammar and thus, the corresponding rules. For example, for symbol No. 6, N_a, test Sentences 1 and 4 (plus others) demonstrate the use of class 1 (Rule 9.1), test Senteces A and 6 (plus others) demonstrate class 2 (Rule 9.3), test sentences 1 and 7 (plus others) demonstrate class 3 (Rule 9.4); the asterisk (*) indicates that the symbol does not have classes 4 through 8.

---

[25] AXD should be omitted.

[26] The use of KY is classical. Modern Hebrew uses SHWA RW&H... as in sentence 26c.

Table 3-7

SENTENCES ILLUSTRATING GRAMMAR SYMBOLS

Symbol No.	Symbol Name	Symbol Class							
		1	2	3	4	5	6	7	8
1	Z	10	*	*	*	*	*	*	*
2	Vgo	6a	*	*	*	*	*	*	*
3	Apa	4,7,8	2	1	*	*	*	*	*
4	Ap	1,2,4,7	*	*	*	*	*	*	*
5	As	A, 23		*	*	*	*	*	*
6	Na	1,4+	A,6+	1,7	*	*	*	*	*
7	Sqo		6a	*	*	*	*	*	*
8	Ns	23a	*	*	*	*	*	*	*
9	Rd		8,9,12		*	*	*	*	*
10	Ro	10,12	*	*	*	*	*	*	*
11	Baa	26b			*	*	*	*	*
12	Bab		*	*	*	*	*	*	*
13	Bac				*	*	*	*	*
14	Bad			*	*	*	*	*	*
15	Bae				*	*	*	*	*
16	Baf				*	*	*	*	*
17	Bba	26b					*	*	*
18	Bbb			*	*	*	*	*	*
19	Ba	26b		*	*	*	*	*	*
20	Bbc			*	*	*	*	*	*
21	Bp	26b		*	*	*	*	*	*
22	Npb	A,2,5+	23	(403)	*	*	*	*	*
23	Npa	A,1,4+	1,2,10+		*	*	*	*	*
24	Npc	A	*	*	*	*	*	*	*

Table 3-7

SENTENCES ILLUSTRATING GRAMMAR SYMBOLS (Continued)

Symbol No.	Symbol Name	Symbol Class							
		1	2	3	4	5	6	7	8
25	Nap	A	1,1a+	10,11a	*	*	*	*	*
26	Np	A,2,5 +	*	*	*	*	*	*	*
27	(Dpa)[27]	*	*	*	*	*	*	*	*
28	(Dpb)[27]	*	*	*	*	*	*	*	*
29	(Dpc)[27]	*	*	*	*	*	*	*	*
30	Dp	6		26a +			*	*	*
31	Ea	A,4c +	*	*	*	*	*	*	*
32	Vb	1,4,5 +	*	*	*	*	*	*	*
33	Vbb	A, 4 +	*	*	*	*	*	*	*
34	Vc			A,2,5 +	*	*	*	*	*
35	Vaa	A,1,2, +	*	*	*	*	*	*	*
36	Va	A,2,3, +	*	*	*	*	*	*	*
37	Bc			*	*	*	*	*	*
38	Xp	A,1,2, +	23		*	*	*	*	*
39	No	A,6,8, +		*	*	*	*	*	*
40	Dpd			*	*	*	*	*	*
41	Rsp	2,4,5, +		*	*	*	*	*	*
42	Nsp	A,1,23 +	2,4,5,			*	*	*	*
43	Nop	5,10,11	A,6,8 +	*		*	*	*	*
44	Nip	10,11,12		*	*	*	*	*	*
45	Npx	2,2a		1		23	*	*	*
46	Vma		6a,10	A,5,6 +	*	*	*	*	*
47	Vmb		12	10,11	*	*	*	*	*
48	Vmc			26a,b,c		*	*	*	*

---

[27] Symbols D_{pa}, D_{pb}, and D_{pc} have been removed from the grammar as a result of modifications made on this project.

Table 3-7 (Continued)
SENTENCES ILLUSTRATING GRAMMAR SYMBOLS

Symbol No.	Symbol Name	Symbol Class							
		1	2	3	4	5	6	7	8
49	Vmd				*	*	*	*	*
50	Vm	1,2,23	3,7	A,5,6+	11,12	A,4,7+	5,7+	26a+	
51	Vmr		6a,10				*	*	*
52	Vmi			11a+	*	*	*	*	*
53	Vp	A,2,5+	*	*	*	*	*	*	*
54	Vrb	6a,10	*	*	*	*	*	*	*
55	Vri	11a,c	*	*	*	*	*	*	*
56	Nv	5,7+			*	*	*	*	*
57	Nw		*	*	*	*	*	*	*
58	Epb			*	*	*	*	*	*
59	Epa	(403)		*	*	*	*	*	*
60	Ep	(403)	*	*	*	*	*	*	*
61	Saa	101c			101a,b	*	*	*	*
62	Sab	A,1,2+	26a+	2b		1e	1ee	*	*
63	Sac	7c,d,f	*	*	*	*	*	*	*
64	Sa	101a	7,23+	A,2,4+	*	*	*	*	*
65	Sro	10	6a,10a	*	*	*	*	*	*
66	Sri	11a	11c	*	*	*	*	*	*
67	Rg	12a	10	11a	*	*	*	*	*
68	Km	26b	26c	*	*	*	*	*	*
69	Kc	7	7g	23	8,9,13	*	*	*	*
70	Kk	7c,d,e	*	*	*	*	*	*	*
71	Ki	1a,7a+	2a	6a	11c	*	*	*	*
72	Kd	26b,c	26a	*	*	*	*	*	*

Table 3-7 (Continued)
SENTENCES ILLUSTRATING GRAMMAR SYMBOLS

Symbol No.	Symbol Name	Symbol Class							
		1	2	3	4	5	6	7	8
73	Sd	7,8,9 +	23	*	*	*	*	*	*
74	S	A,1,2 +	7,8,9,+	7,c,d,e	*	*	*	*	*
75	S_i	10,2a +	7a	7a	*	*	*	*	*
76	S_c	A,1,2 +	10,7a +	1b	*	*	*	*	*
77	A	1,2,4 +	*	*	*	*	*	*	*
78	B	26a						*	*
79	C	A		26b	7	7f	23		6
80	D	6	1,23	1,23			5	*	*
81	E	A,4b +	*	*	*	*	*	*	*
82	G		*	*	*	*	*	*	*
83	H	26a +	*	*	*	*	*	*	*
84	I							*	*
85	J	1,2,11 +	*	*	*	*	*	*	*
86	L	7e	1e			*	*	*	*
87	N	1,2,4 +	1,3,7 +	1,1a +	*	*	*	*	*
88	O	A,26a	*	*	*	*	*	*	*
89	P	A,8,13 +	*	*	*	*	*	*	*
90	Q	1a,7a +	7aa	*	*	*	*	*	*
91	R	8,9,12	2,4,5 +	10,23 +	12a,26c	2a,6a	*	*	*
92	T	26a	26a	A,7	26a	1a,7a +	A,1,2 +	1b	
93	U	A		1d	7c	7d	7e,f		*
94	V	A,1,2,+	*	*	*	*	*	*	*
95	W		*	*	*	*	*	*	*
96	Y	5,7 +	*	*	*	*	*	*	*

Tree diagrams of the test sentences are contained in Section 2.3.1 of Part II of this report which show the specified symbol as it relates to the structure of the sentence. Sentence numbers enclosed in parentheses refer to tree diagrams in Section 2.3.2 of Part II of this report. In such cases, the symbol was tested by the algorithm for analyzing Hebrew sentences (Part IV of this report) but not by the present algorithm. For example, Symbol No. 60, E_p, Class 1 refers to sentence (403). Test sentence (403) is the analysis of Sentence 4c. The analysis algorithm computed a different (but equally valid) parsing of the sentence which included Symbols E_p and E_{pa} which had not been previously tested by the present algorithm.

3.5 Conclusion

Tests of the algorithm have fully demonstrated its capability for generating sentences in modern Hebrew. Even though all of the grammar rules have not been tested, experience has shown that when errors are made they are due to the rules of the grammar and not to the logic of the algorithm. It is reasonable to conclude, therefore, that the algorithm is essentially complete.

Two aspects of the algorithm need further attention. First, there are several computations of the algorithm that are dependent upon the "content" of the grammar upon which it operates. If the algorithm were modified to free these computations from such dependence, a modification which is feasible, the algorithm could be used to generate sentences in other Semitic languages, such as Arabic, whenever grammars of these languages become available. It could also be used to aid in the development of such grammars.

The second aspect of the algorithm needing further attention is the format of the sentence description input data prepared by the user. The input to the algorithm is the equivalent of the output of an English-to-Hebrew transfer grammar.[28] As such, the format is complex and the manual preparation of the data is cumbersome. This can be tolerated for research purposes, but if more extensive use is made of the algorithm, consideration should be given to providing automatic aids for preparing the input data.

---

[28] See Section 2.1.2 (Part II) for more detail.

Appendix

PART III

APPENDIX A

LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX FOR SYNTHESIS OF SENTENCES

```
        LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z)                  RULE NO.


      F-(1F 00CL1DNGPRAVIT SWWWW 0 0) =               (  2. 1)
      L-(10 001L0000000000 00000 0 0) +
      F-(1F 00CL0DNGPRAVIT SWWWW 0 0)


      F-(11 21CLYDNGPRAVIT SWWWW 0 0) =               (  3. 1)
      F-(10 00CLYDNGPRAVIT SWWWW 0 0) +
      C-(10 00100000000000 00000 0 0) +
      F-(10 00CLYDNGPRAVIT SWWWW 0 0)


      F-(11 22CLYDNGPRAVIT SWWWW 0 0) =               (  3. 2)
      F-(10 00CLYDNGPRAVIT SWWWW 0 0) +
      C-(10 00200000000000 00000 0 0) +
      F-(10 00CLYDNGPRAVIT SWWWW 0 0)


      F-(11 KBCLYDNGPRAVIT SWWWW 6 4) =               (  3. 3)
      F-(10 00CLYDNGPRAVIT SWWWW 0 0) +
      I-(10 00300000000000 00000 0 0) +
      F-(11-1BCLYDNGPRAVIT SWWWW 0 0)


      F-(12 21CLYDNGPRAVIT SWWWW 0 0) =               (  3. 4)
      F-(10 00CLYD999RAVIT SWWWW 0 0) +
      C-(10 00100000000000 00000 0 0) +
      F-(10 00CLYD999RAVIT SWWWW 0 0)


      F-(12 22CLYDNGPRAVIT SWWWW 0 0) =               (  3. 5)
      F-(10 00CLYD999RAVIT SWWWW 0 0) +
      C-(10 00200000000000 00000 0 0) +
      F-(10 00CLYD999RAVIT SWWWW 0 0)
      LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z)            RULE NO.


    F-(12 KBCLYDNGPRAVIT SWWWW 6 4) =           (  3. 6)
    F-(10 00CLYD999RAVIT SWWWW 0 0) +
    T-(10 0030000000000 00000 0 0) +
    F-(12-1BCLYD999RAVIT SWWWW 0 0)


    F-(13 KBCLYDNGPRAVIT SWWWW 6 4) =           (  3. 7)
    F-(10 00CLYDNGPRAVIT SWWWW 0 0) +
    T-(10 0030000000000 00000 0 0) +
    F-(13-1BCLYDNGPRAVIT SWWWW 0 0)


    F-(13 2BCLYDNGPRAVIT SWWWW 0 0) =           (  3. 8)
    F-(10 00CLYDMGPRAVIT SWWWW 0 0) +
    T-(10 0030000000000 00000 0 0) +
    F-(10 00CLYDNGPRAVIT SWWWW 0 0)


    F-(14 KBCLYDNGPRAVIT SWWWW 6 4) =           (  3. 9)
    F-(10 00CLYDNGPRAVIT SWWWW 0 0) +
    F-(14-10CLYDNGPRAVIT SWWWW 0 0)


    F-(14 2BCLYDNGPRAVIT SWWWW 0 0) =           (  3.10)
    F-(10 00CLYDNGPRAVIT SWWWW 0 0) +
    F-(10 00CLYDNGPRAVIT SWWWW 0 0)


    Z-(10 0010Y0000R0000 00000 0 0) =           (  4. 1)
    P-(10 0011Y0000R0000 00000 0 0) +
    R-(10 0030009990000 00000 0 0)
   VQO(10 0010YONGPR3VIT 00000 0 0) +
    VP(10 0010YONGPR2VIT 00000 0 0)
```

104

```
        LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z)             RULE NO.


   VGO(10 0010Y0NGPR4VIT 00000 0 0) =            (  5. 1)
    VP(10 0010Y0NGPR5VIT 00000 0 0)


   VGO(10 0010Y0NGPR8VIT 00000 0 0) =            (  5. 2)
    VP(10 0010Y0NGPR3VIT 00000 0 0)


   APA(10 0010YDNG000000 00000 0 0) =            (  6. 1)
    A-(10 0011YDNG000000 99999 0 0) +
    D-(94 9060000000000 99999 0 0)


   APA(10 0020YDNG000000 00000 0 0) =            (  6. 2)
    A-(10 0011Y1NG000000 99999 0 0) +
    XP(11 9990000090000 00000 913)


   APA(10 0030YDNG000000 00000 0 0) =            (  6. 3)
    A-(10 0011Y2NG000000 99999 0 0) +
    XP(11 9990000020000 00000 0 0)


    AP(10 0010YDNG000000 00000 210) =            (  7. 1)
   APA(10 0090YDNG000000 00000 210)


    AP(10 0010YD2G000000 00000 0 0) =            (  7. 2)
   APA(10 0090YD3G000000 00000 0 0)


    AS(10 0010000000000 00000 0 0) =             (  8. 1)
    U-(10 0010000000000 00000 0 0) +
    R-(10 0030099900000 00000 0 0)
```

LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

```
SYMBOL(MF/KBCLYONGPRAVIT/S/-w-/X/Z)              RULE NO.


     AS(10 00200000000000 00000 0 0) =          (  8. 2)
     J-(10 00100000000000 00000 0 0) +
     NP(10 00100999900000 00000 0 0)


     NA(10 0010YONGP00000 00000 2 9) =          (  9. 1)
     N-(10 0011YONGP00000 99999 2 9)


     NA(10 0010Y1NGP00000 00000 0 0) =          (  9. 2)
     N-(10 0011YONGP00000 99999 0 0) +
     E-(10 00100ONG000000 00000 0 0)


     NA(10 0020Y2NGP00000 00000 0 0) =          (  9. 3)
     H-(10 0011Y000000000 00000 0 0) +
     N-(10 001000NGP00000 99999 0 0)


     NA(10 0030Y2NGP00000 00000 0 0) =          (  9. 4)
     N-(14 9991YONGP00000 99999 1 6)


    SGO(10 0010YONGPR0000 00000 0 0) =          ( 10. 1)
    VGO(11 9910YONGPR9999 00000 0 0)


    SGO(10 0020YONGPR0000 00000 0 0) =          ( 10. 2)
    NSP(12 999099NGP00000 00000 0 0) +
    VGO(11 9910YONGPR9999 00000 0 0)


     NS(10 0010Y2NGP00000 00000 0 0) =          ( 11. 1)
     J-(10 0011YONGP00000 99999 0 0) +
     R-(10 00300099900000 00000 0 0)
```

LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

```
SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z)           RULE NO.


     RD(10 0010YDNGP00000 00000 0 0) =          ( 12. 1)
     R-(10 0011Y0NG300000 00000 0 0)


     RD(10 0020Y2NG300000 00000 0 0) =          ( 12. 2)
     H-(10 0011Y000000000 00000 0 0) +
     R-(10 001000NG300000 00000 0 0)


     RD(10 0030Y2NG300000 00000 0 0) =          ( 12. 3)
     H-(10 0011Y000000000 00000 0 0) +
     R-(10 002000NG300000 00000 0 0)


     RO(10 0010Y2NGP00000 00000 0 0) =          ( 13. 1)
     O-(10 0011Y000000000 00000 0 0) +
     R-(10 003000NGP00000 00000 0 0)


    BAA(10 0010001G000000 00000 0 0) =          ( 14. 1)
     B-(10 0010001G000000 00000 0 0)


    BAA(10 0020002G000000 00000 0 0) =          ( 14. 2)
     B-(10 0020002G000000 00000 0 0)


    BAA(10 00300031000000 00000 0 0) =          ( 14. 3)
     B-(10 00300012000000 99999 0 0)


    BAA(10 00300032000000 00000 0 0) =          ( 14. 4)
     B-(10 00300011000000 99999 0 0)
```

```
LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z)              RULE NO.


   BAB(10 00100031000000 00000 0 0) =            ( 15. 1)
    B-(10 00400012000000 00000 0 0)


   BAB(10 00100032000000 00000 0 0) =            ( 15. 2)
    B-(10 00400011000000 00000 0 0)


   BAC(10 0010003G000000 00000 0 0) =            ( 16. 1)
    B-(10 0010001G000000 00000 0 0) +
    B-(10 0040001G000000 00000 0 0)


   BAC(10 0020003G000000 00000 0 0) =            ( 16. 2)
    B-(10 0020003G000000 00000 0 0) +
    B-(10 0040001G000000 00000 0 0)


   BAC(10 0020003G000000 00000 0 0) =            ( 16. 3)
    1-(10 0020003G000000 00000 0 0) +
    B-(10 0040001G000000 00000 0 0)


   BAC(10 00300031000000 00000 0 0) =            ( 16. 4)
    B-(10 00300012000000 99999 0 0) +
    B-(10 00400011000000 00000 0 0)


   BAC(10 00300032000000 00000 0 0) =            ( 16. 5)
    B-(10 00300011000000 99999 0 0) +
    B-(10 00400012000000 00000 0 0)


   BAD(10 0010003G000000 00000 0 0) =            ( 17. 1)
    B-(10 00400031000000 00000 0 0)
```

107

LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

```
SYMBOL(MF/KBCLYONGPRAVIT/S/-W-/X/Z)            RULE NO.


   BAD(10 0020003G000000 00000 0 0) =              ( 17. 2)
    B-(10 0030003100000 99999 0 0)


   BAE(10 0010003G000000 00000 0 0) =              ( 18. 1)
    B-(10 0050001200000 00000 0 0)


   BAE(10 0020003G000000 00000 0 0) =              ( 18. 2)
    B-(10 0050002200000 00000 0 0)


   BAE(10 0020003G000000 00000 0 0) =              ( 18. 3)
    I-(10 0020003200000 00000 0 0) +
    B-(10 0050003200000 00000 0 0)


   BAE(10 0030003G000000 00000 0 0) =              ( 18. 4)
    I-(10 0030001100000 99999 0 0) +
    B-(10 0050003200000 00000 0 0)


   BAF(10 0010003G000000 00000 0 0) =              ( 19. 1)
    B-(10 0060001100000 00000 0 0)


   BAF(10 0020003G000000 00000 0 0) =              ( 19. 2)
    B-(10 0060002100000 00000 0 0)


   BAF(10 0020003G000000 00000 0 0) =              ( 19. 3)
    I-(10 0020003100000 00000 0 0) +
    B-(10 0060003100000 00000 0 0)
```

```
LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYBNGPRAVIT/S/-W-/X/Z)          RULE NO.


   BAF(10 0030003G000000 00000 0 0) =           ( 19. 4)
    I-(10 00300031000000 99999 0 0) +
    B-(10 00000031000000 00000 0 0)


   BBA(10 001000NG000000 00000 0 0) =           ( 20. 1)
   BAA(10 009000NG000000 00000 0 0)


   BBA(10 0020003G000000 00000 0 0) =           ( 20. 2)
   BAB(10 0010003G000000 00000 0 0)


   BBA(10 0030003G000000 00000 0 0) =           ( 20. 3)
   BAC(10 0090003G000000 00000 0 0)


   BBA(10 0040003G000000 00000 0 0) =           ( 20. 4)
   BAD(10 0090003G000000 00000 0 0)


   BBA(10 0050003G000000 00000 0 0) =           ( 20. 5)
   BAD(10 0090003G000000 00000 0 0) +
    C-(10 00100000000000 00000 0 0) +
   BAA(10 0090009G000000 99999 0 0)


   BBB(10 0010003G000000 00000 0 0) =           ( 21. 1)
   BAE(10 0090003G000000 00000 0 0)


   BBB(10 0020003G000000 00000 0 0) =           ( 21. 2)
   BAE(10 0090003G000000 00000 0 0) +
    C-(10 00100000000000 00000 0 0) +
   BBA(10 0090009G000000 00000 0 0)
```

```
      LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z)                   RULE NO.


     BA(10 001000NG000000 00000 0 0) =                ( 22. 1)
    BBA(10 009000NG000000 00000 0 0)


     BA(10 0020003G000000 00000 0 0) =                ( 22. 2)
    BBB(10 0090003G000000 00000 0 0)


    BBC(10 0010003G000000 00000 0 0) =                ( 23. 1)
    BAF(10 0090003G000000 00000 0 0)


    BBC(10 0020003G000000 00000 0 0) =                ( 23. 2)
    BAF(10 0090003G000000 00000 0 0) +
     C-(10 00100000000000 00000 0 0) +
     BA(10 0090003G000000 00000 0 0)


     BP(10 001000NG000000 00000 0 0) =                ( 24. 1)
     BA(10 009000NG000000 00000 0 0)


     BP(10 0020003G000000 00000 0 0) =                ( 24. 2)
    BBC(10 0090003G000000 00000 0 0)


    NPB(10 0010YDNGP00000 00000 0 0) =                ( 25. 1)
     NA(12 9990YDNGP00000 00000 0 0) +
     AP(91 99909DNG000000 00000 0 0) +
     RD(90 00900DNGP00000 00000 0 0) +
     AS(91 99900000000000 00000 0 0)


    NPB(10 0020Y2NGP00000 00000 0 0) =                ( 25. 2)
     NS(10 0010Y2NGP00000 00000 0 0) +
     AP(91 999092NG000000 00000 0 0)
```

```
        LISTING OF THE GRAMMAR RUL_S OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z)               RULE NO.


   NPB(10 0030YDNGP00000 00000 0 0) =              ( 25. 3)
    EP(10 0010YDNGP00000 00000 0 0)


   NPA(10 00100DNGP00000 00000 0 0) =              ( 26. 1)
    BP(90 009000NG000000 00000 0 0) +
   NPB(10 00900DNGP00000 00000 0 0) +
    D-(94 99200000000000 99999 0 0)


   NPA(10 00200DNGP00000 00000 0 0) =              ( 26. 2)
    BP(90 009000NG000000 00000 0 0) +
    D-(10 001000NGP00000 99999 0 0) +
    D-(94 99100099900000 99999 0 0) +
   NPB(10 00900D99900000 00000 0 0) +
    D-(94 99200000000000 99999 0 0)


   NPA(10 00300DNGP00000 00000 0 0) =              ( 26. 3)
    BC(10 00900DNG000000 00000 0 0)


   NPC(10 00100DNGP00000 00000 0 0) =              ( 27. 1)
   NPA(12 99919DNGP00000 00000 0 0)


   NAP(10 00100DNGP00000 00000 0 0) =              ( 28. 1)
    T-(10 00300000000000 00000 0 0) +
   NPC(13 99119DNCP00000 00000 0 0) +
    T-(10 00300000000000 00000 0 0)


   NAP(10 00200DNGP00000 00000 0 0) =              ( 28. 2)
    XP(14 99909000090000 00000 0 0)
```

```
        LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z)               RULE NO.


   NAP(10 00300DNGP00000 00000 0 0) =               ( 28. 3)
    RG(11 99909DNGP00000 00000 0 0)


    NP(10 00100DNGP00000 00000 0 0) =               ( 29. 1)
   NPA(10 00900DNGP00000 00000 0 0) +
   NAP(90 009002NGP00000 00000 0 0)


   DPA(10 0010000000000 00000 0 0) =                ( 30. 1)
    D-(10 0031900000000 99999 0 0) +
    BP(90 0090009900000 00000 0 0)


   DPA(10 0020000000000 00000 0 0) =                ( 30. 2)
    BP(10 0090009900000 00000 0 0) +
    D-(10 0031900000000 99999 0 0)


   DPB(10 0010000000000 00000 0 0) =                ( 31. 1)
    D-(10 0041900000000 99999 0 0) +
    D-(90 0060000000000 99999 0 0)


   DPC(10 0010000000000 00000 0 0) =                ( 32. 1)
    D-(10 0051900000000 99999 0 0) +
    D-(90 0060000000000 99999 0 0)


    DP(10 0010000000000 00000 0 0) =                ( 33. 1)
    D-(11 9911900000000 99999 0 0)


    DP(10 0020000000000 00000 0 0) =                ( 33. 2)
    D-(11 9921900000000 99999 0 0)
```

```
      LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYONGPRAVIT/S/-W-/X/Z)            RULE NO.


    DP(10 00300000000000 00000 0 0) =              ( 33. 3)
   DPA(11 99119000000000 00000 0 0)


    DP(10 00400000000000 00000 0 0) =              ( 33. 4)
   DPB(11 99119000000000 00000 0 0)


    DP(10 00500000000000 00000 0 0) =              ( 33. 5)
   DPC(11 99119000000000 00000 0 0)


    DP(10 00600000000000 00000 0 0) =              ( 33. 6)
    D-(10 00619000000000 99999 0 0) +
    D-(90 00600000000000 99999 0 0)


    DP(10 00700000000000 00000 0 0) =              ( 33. 7)
    XP(11 9990Y000090000 00000 0 0)


    EA(10 0010YONGPRAV00 SWWWW 2 9) =              ( 34. 1)
    E-(10 0011YONGPRAV00 SWWWW 2 9)


    EA(10 0010Y2NGPRAV00 SWWWW 0 0) =              ( 34. 2)
    H-(10 0011Y000000000 00000 0 0) +
    E-(10 001000NGPRAV00 SWWWW 0 0)


    VB(10 0010YONGPRAVIT SWWWW 0 0) =              ( 35. 1)
    V-(10 0011YONGPRAVIT SWWWW 0 0)
```

```
        LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

SYMBOL(MF/KPCLYONGPRAVIT/S/-w-/X/Z)             RULE NO.


     VB(10 0010YONG2RAV22 SWWWW 0 0) =             ( 35. 2)
     V-(10 0011YONG2RAV22 SWWWW 0 0)


     VB(10 0010YONG2RAV22 SWWWW 0 0) =             ( 35. 3)
     W-(10 0011Y0000RAV00 SWWWW 0 0)


     VB(10 0010YONGPRAV32 SWWWW 0 0) =             ( 35. 4)
     V-(10 0014YONGPRAV12 SWWWW 0 0) +
     U-(10 00200000000000 00000 0 0)


     VB(10 001000NGPRAV32 SWWWW 0 0) =             ( 35. 5)
     V-(10 001000NGPRAV22 SWWWW 0 0) +
     U-(10 00200000000000 00000 0 0)


     VB(10 001010NGPRAV32 SWWWW 0 0) =             ( 35. 6)
     V-(10 001410NGPRAV22 SWWWW 0 0)


    VBB(10 0010YONGPRAVIT SWWWW 5 7) =             ( 36. 1)
     VB(10 0010YONGPRAVIT SWWWW 517)


    VBB(10 0010YONGPRAVI3 SWWWW 0 0) =             ( 36. 2)
     EA(10 0010Y1NGPRAV00 SWWWW 0 0)


    VBB(10 002000NGPR1113 SWWWW 0 0) =             ( 36. 3)
     EA(90 001000NGPR1100 SWWWW 0 0)
```

```
      LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYONGPRAVIT/S/-W-/X/Z)          RULE NO.


   VBB(10 003000NG3R1113 SWWWW 0 0) =          ( 36. 4)
    R-(10 002000NG300000 00000 0 0)


    VC(10 0010YONGPRAVIT SWWWW1117) =          ( 37. 1)
    W-(10 001000000RAV00 SWWWW 0 0) +
   VBB(10 0090YONGPRAVIT SWWWW1117)


    VC(10 0020YONGPRAVIT SWWWW1117) =          ( 37. 2)
   VBB(10 0090YONGPRAVIT SWWWW1117) +
    W-(10 001000000RAV00 SWWWW 0 0)


    VC(10 0030YONGPRAVIT SWWWW1117) =          ( 37. 3)
   VBB(10 0090YONGPRAVIT SWWWW1117)


   VIA(10 0010YONGPRAVIT SWWWW1117) =          ( 38. 1)
    VC(10 0090YONGPRAVIT SWWWW1117)


   VIA(10 0010YONGPRAVI4 SWWWW 0 0) =          ( 38. 2)
    V-(10 0011YONGP01VI1 00000 0 0) +
    VC(10 009000NGPRAVI3 SWWWW 0 0)


   VIA(10 0010YONGPRAVI5 SWWWW 0 0) =          ( 38. 3)
    V-(10 0011YONGP01VI2 00000 0 0) +
    VC(10 009000NGPRAVI3 SWWWW 0 0)


   VIA(10 0010YONGPRAVI6 SWWWW 0 0) =          ( 38. 4)
    D-(10 0071Y000000000 00000 0 0) +
    VC(10 009000NGPRAVI1 SWWWW 0 0)
```

```
         LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

      SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z)               RULE NO.


        VIA(10 0010YONGPRAVI7 SWWWW 0 0) =               ( 38. 5)
         D-(10 0081Y00000000 00000 0 0) +
         VC(10 009000NGPRAVI2 SWWWW 0 0)


         VA(10 0010YONGPRAVIT SWWWW 210) =               ( 39. 1)
        VIA(10 0010YONGPRAVIT SWWWW 210)


         VA(10 0010Y02GPRAVIT SWWWW 0 0) =               ( 39. 2)
        VIA(10 0010Y03GPRAVIT SWWWW 0 0)


         BC(10 00100DNG000000 00000 4 9) =               ( 40. 1)
         BP(10 009000NG000000 00000 0 0)


         BC(10 002002NG000000 00000 0 0) =               ( 40. 2)
         H-(10 00100000000000 00000 0 0) +
         BP(10 009000NG000000 00000 0 0)


         XP(10 0010Y0000R0000 00000 0 0) =               ( 41. 1)
         P-(10 0011Y0000R0000 00000 0 0) +
         NP(11 99100999900000 00000 0 0)


         XP(10 0010Y00001000 00000 0 0) =                ( 41. 2)
         NA(10 0091Y99990000 00000 0 0) +
         U-(10 00700000000000 00000 0 0)


         XP(10 0020Y0000R0000 00000 0 0) =               ( 41. 3)
         Z-(10 0010Y000R0000 00000 0 0)
```

LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

```
SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z)              RULE NO.


   XP(10 0030Y0000R0000 00000 0 0) =              ( 41. 4)
   P-(10 0011Y0000R0000 00000 0 0) +
   RG(11 99900999900000 00000 0 0)


   NO(10 0010Y200000000 00000 0 0) =              ( 42. 1)
   G-(10 0011Y000000000 00000 0 0) +
   NP(10 00100299900000 00000 0 0)


   NO(10 0020Y200000000 00000 0 0) =              ( 42. 2)
   G-(10 0011Y000000000 00000 0 0) +
   RG(10 00900299900000 00000 0 0)


  DPD(10 0010Y000000000 00000 0 0) =              ( 43. 1)
   D-(11 9911Y000000000 99999 0 0)


  DPD(10 0020Y000000000 00000 0 0) =              ( 43. 2)
   D-(11 9921Y000000000 99999 0 0)


  RSP(10 001002NGP00000 00000 0 0) =              ( 44. 1)
   R-(10 002000NGP00000 00000 0 0) +
  NPC(93 991002NGP00000 00000 0 0)


  RSP(10 002002NGP00000 00000 0 0) =              ( 44. 2)
   R-(12 991000NGP00000 00000 0 0)


  NSP(10 00100DNGP00000 00000 0 0) =              ( 45. 1)
   NP(12 99100DNGP00000 00000 0 0)
```

LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

```
SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z)                 RULE NO.


      NSP(10 00200DNGP00000 00000 0 0) =              ( 45. 2)
      RSP(12 999002NGP00000 00000 0 0)


      NSP(10 00300211300000 00000 0 0) =              ( 45. 3)
       KN(10 00900000000000 00000 0 0)


      NSP(10 00400D11300000 00000 0 0) =              ( 45. 4)
       NV(10 00900000099000 00000 0 0)


      NOP(10 0010YD00000000 00000 2 9) =              ( 46. 1)
       NP(11 9910YD99900000 00000 2 9)


      NCP(10 0020Y200000000 00000 0 0) =              ( 46. 2)
       NO(11 9990Y200000000 00000 0 0)


      NIP(10 0010Y0000R0000 00000 0 0) =              ( 47. 1)
       XP(11 9990Y0000R0000 00000 0 0)


      NIP(10 0020Y0000P0000 00000 0 0) =              ( 47. 2)
       NV(11 9990Y000099000 00000 0 0)


      NPX(10 0010YDNGP00000 00000 0 0) =              ( 48. 1)
       AP(11 9910Y1NGP00000 00000 0 0)


      NPX(10 0020YDNGP00000 00000 0 0) =              ( 48. 2)
      DPB(14 9910Y00C000000 00000 0 0)
      LISTING OF THE GRAMMAR RUL S OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z)            RULE NO.


     NPX(10 0030YDNGP00000 00000 0 0) =           ( 48. 3)
      NP(12 9910YDNGP00000 00000 0 0)


     NPX(10 0040YDNGP00000 00000 0 0) =           ( 48. 4)
     RSP(12 9990Y2NGP00000 00000 0 0)


     NPX(10 0050YDNGP00000 00000 0 0) =           ( 48. 5)
      XP(11 9990Y000090000 00000 0 0)


     NPX(10 0060YDNGF00000 00000 0 0) =           ( 48. 6)
      RC(11 9990Y000000000 00000 0 0)


     VMA(10 00100000003000 00000 0 0) =           ( 49. 1)
      R-(10 00300099900000 00000 0 0) +
      DP(94 99909000000000 00000 0 0) +
      Z-(91 99909000090000 00000 0 0) +
      DP(94 99909000000000 00000 0 0)


     VMA(10 00200000003000 00000 0 0) =           ( 49. 2)
      DP(94 99909000000000 00000 0 0) +
      RO(10 00109299900000 00000 0 0) +
      Z-(91 99909000090000 00000 0 0) +
      DP(94 99909000000000 00000 0 0)


     VMA(10 00300000003000 00000 0 0) =           ( 49. 3)
      DP(94 99909000000000 00000 0 0) +
      Z-(91 99909000090000 00000 0 0) +
     NOP(11 99909900000000 00000 0 0) +
      DP(94 99909000000000 00000 0 0)
```

LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

```
SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z)              RULE NO.


   VMB(10 001000000R4000 00000 0 0) =            ( 50. 1)
    R-(10 00300099900000 00000 0 0) +
    DP(94 99909000000000 00000 0 0) +
   NIP(11 999090000R0000 00000 0 0) +
    DP(94 99909000000000 00000 0 0)


   VMB(10 002000000R4000 00000 0 0) =            ( 50. 2)
    DP(94 99909000000000 00000 0 0) +
    RO(11 99109299900000 00000 0 0) +
   NIP(11 999090000R0000 00000 0 0) +
    DP(94 99909000000000 00000 0 0)


   VMB(10 003000000R4000 00000 0 0) =            ( 50. 3)
    DP(94 99909000000000 00000 0 0) +
   NOP(11 99909900000000 00000 0 0) +
   NIP(11 999090000R0000 00000 0 0) +
    DP(94 99909000000000 00000 0 0)


   VMC(10 00100000007000 00000 0 0) =            ( 51. 1)
    R-(10 00300099900000 00000 0 0) +
    Z-(91 99909000090000 00000 0 0) +
    DP(94 99909000000000 00000 0 0) +
    KD(11 99909000000000 00000 0 0)


   VMC(10 00200000000700 00000 0 0) =            ( 51. 2)
    DP(94 99909000000000 00000 0 0) +
    RO(19 99109299900000 00000 0 0) +
    Z-(91 99909000090000 00000 0 0) +
    DP(94 99909000000000 00000 0 0) +
    KD(11 99909000000000 00000 0 0)
```

```
      LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z)            RULE NO.


   VMC(10 0030000007000 00000 0 0) =          ( 51. 3)
    DP(94 9990900000000 00000 0 0) +
    Z-(91 9990900090000 00000 0 0) +
   NOP(91 9990990000000 00000 0 0) +
    DP(94 9990900000000 00000 0 0) +
    KD(11 9990900000000 00000 0 0)


   VMD(10 0010000008000 00000 0 0) =          ( 52. 1)
    K-(10 0030099900000 00000 0 0) +
    DP(94 9990900000000 00000 0 0) +
    Z-(91 9990900090000 00000 0 0) +
    NP(11 9910999900000 00000 0 0) +
    DP(94 9990900000000 00000 0 0)


   VMD(10 0020000008000 00000 0 0) =          ( 52. 2)
    DP(94 9990900000000 00000 0 0) +
    RO(11 9910929990000 00000 0 0) +
    Z-(91 9990900090000 00000 0 0) +
    NP(11 9910999900000 00000 0 0) +
    DP(94 9990900000000 00000 0 0)


   VMD(10 0030000008000 00000 0 0) =          ( 52. 3)
    DP(94 9990900000000 00000 0 0) +
    Z-(91 9990900090000 00000 0 0) +
   NCP(11 9990990000000 00000 0 0) +
    NP(11 9910999900000 00000 0 0) +
    DP(94 9990900000000 00000 0 0)


    VM(10 001000NGP01000 00000 0 0) =         ( 53. 1)
    DP(94 9990900000000 00000 0 0) +
   NPX(11 999099NGP00000 00000 0 0) +
    DP(94 9990900000000 00000 0 0)
```

LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

```
SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z)                RULE NO.


     VM(10 002000NGP02000 00000 0 0) =              ( 53. 2)
     DP(94 99909000000000 00000 0 0)


     VM(10 003000NGP03000 00000 0 0) =              ( 53. 3)
    VMA(10 00900000003000 00000 0 0)


     VM(10 004000NGPR4000 00000 0 0) =              ( 53. 4)
    VMB(10 009000000R4000 00000 0 0)


     VM(10 005000NGPR5000 00000 0 0) =              ( 53. 5)
     DP(94 99909000000000 00000 0 0) +
     XP(11 999090000R0000 00000 0 0) +
     DP(94 99909000000000 00000 0 0)


     VM(10 006000NGP06000 00000 0 0) =              ( 53. 6)
     DP(94 99909000000000 00000 0 0) +
     P-(10 00100000010000 00000 0 0) +
     NV(10 00900000099000 00000 0 0) +
     DP(94 99909000000000 00000 0 0)


     VM(10 007000NGP07000 00000 0 0) =              ( 53. 7)
    VMC(10 00900000007000 00000 0 0)


     VM(10 008000NGP08000 00000 0 0) =              ( 53. 8)
    VMD(10 00900000008000 00000 0 0)


    VMR(10 001000NGP01000 00000 0 0) =              ( 54. 1)
     DP(94 99909000000000 00000 0 0)
```

LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

```
SYMBOL(MF/KBCLYDNGPRAVIT/S/-w-/X/Z)                RULE NO.


     VMR(10 002000NGP03000 00000 0 0) =             ( 54. 2)
     VMA(10 00900000003000 00000 5 6)


     VMR(10 003000NGP04000 00000 0 0) =             ( 54. 3)
     VMB(10 00900000004000 00000 5 6)


     VMR(10 004000NGPR5000 00000 0 0) =             ( 54. 4)
      DP(94 99909000000000 00000 0 0) +
      Z-(11 991090000R0000 00000 0 0) +
      DP(94 99909000000000 00000 0 0)


     VMR(10 005000NGP06000 00000 0 0) =             ( 54. 5)
     VMD(10 00900000006000 00000 5 6)


     VMI(10 001000NGPR4000 00000 0 0) =             ( 55. 1)
      R-(10 00300099900000 00000 0 0) +
      DP(94 99909000000000 00000 0 0) +
      Z-(10 001000000R0000 00000 0 0) +
      DP(94 99909000000000 00000 0 0)


     VMI(10 002000NGPR4000 00000 0 0) =             ( 55. 2)
      DP(94 99909000000000 00000 0 0) +
      R0(11 99109299900000 00000 0 0) +
      Z-(10 001000000F0000 00000 0 0) +
      DP(94 99909000000000 00000 0 0)


     VMI(10 003000NGPR4000 00000 0 0) =             ( 55. 3)
      DP(94 99909000000000 00000 0 0) +
      Z-(10 001000000R0000 00000 0 0) +
     NOP(11 99909900000000 00000 0 0) +
      DP(94 99909000000000 00000 0 0)
```

```
   LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYONGPRAVIT/S/-W-/X/Z)                RULE NO.


      VP(10 0010YONGP01VIT 00000 0 0) =             ( 56. 1)
      UP(94 99909000000000 00000 0 0) +
      VA(10 0010YONGP01VIT 99999 0 0) +
      VM(12 99900ONGP01000 00000 0 0)


      VP(10 0010YONGPRAVIT 00000 114) =             ( 56. 2)
      DP(94 99909000000000 00000 0 0) +
      VA(10 0010YONGPRAVIT 99999 114) +
      VM(11 999000999RA000 00000 114)


     VRB(10 0010YONGPRAVIT 00000 0 0) =             ( 57. 1)
      VA(10 0010YONGPRAVIT 99999 0 0) +
     VMR(11 999090999RA000 00000 0 0)


     VRI(10 0010YONGPR4VIT 00000 0 0) =             ( 58. 1)
      VA(10 0010YONGPR4VIT 99999 0 0) +
     VMI(11 999090000R4000 00000 0 0)


      NV(10 0010Y0000RA000 00000 0 0) =             ( 59. 1)
      Y-(10 0013Y0000RA000 99999 0 0) +
      VM(11 999090999RA000 00000 0 0)


      NV(10 0020Y0000RA000 00000 0 0) =             ( 59. 2)
      Y-(10 0013Y0000RA000 99999 0 0) +
      R-(10 00300099900000 00000 0 0) +
      VM(11 999090999RA000 00000 0 0)


      NV(10 0030Y0000RA000 00000 0 0) =             ( 59. 3)
      Y-(10 0013Y0000RA000 99999 0 0) +
     NSP(11 99909999900000 00000 0 0) +
      VM(11 999090999RA000 00000 0 0)
```

```
    LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYONGPRAVIT/S/-W-/X/Z)                 RULE NO.


   VW(10 0010Y0000RA000 00000 0 0) =               ( 60. 1)
   W-(11 9913Y0000RA000 99999 0 0) +
   VM(10 0090000000RA000 00000 0 0)


  EPB(10 0010YONGPR3100 00000 0 0) =               ( 61. 1)
   G-(11 9912YONGP03100 99999 0 0) +
  NPA(11 99909099900000 00000 0 0)


  EPB(10 0020Y2NGPR3100 00000 0 0) =               ( 61. 2)
   G-(10 0012YONGP03100 99999 0 0) +
   K-(10 00300099900000 00000 0 0)


  EPA(10 0010YONGPRAV00 00000 0 0) =               ( 62. 1)
   EA(10 0012YONGPRAV00 99999 0 0) +
   VM(11 999090999RA000 00000 0 0)


  EPA(10 0020YONGPR3100 00000 0 0) =               ( 62. 2)
  EPB(10 0090YONGPR3100 00000 0 0)


   EP(10 0010YONGP00000 00000 2 0) =               ( 63. 1)
  EPA(10 0090YONGP99900 00000 210)


   EP(10 0010YD2GP00000 00000 0 0) =               ( 63. 2)
  EPA(10 0010YD3GP99900 00000 0 0)


  SAA(10 0010YONGP00VIT 00000 317) =               ( 64. 1)
   XP(11 99900000010000 00000 0 0) +
   VA(10 0010YONGP01VIT 99999 0 0) +
  NSP(12 999099NGP00000 00000 0 0) +
   DP(94 99909000000000 00000 0 0)
   LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYONGPRAVIT/S/-W-/X/Z)          RULE NO.


    SAA(10 0020YONGP00VIT 00000 317) =          ( 64. 2)
    NSP(12 999099NGP00000 00000 0 0) +
     VA(10 0010YONGP01VIT 99999 0 0) +
     XP(11 99900000010000 00000 0 0) +
     DP(94 99909000000000 00000 0 0)


    SAA(10 0030YONGP00VIT 00000 317) =          ( 64. 3)
     VA(10 0010YONGP01VIT 99999 0 0) +
     XP(11 99900000010000 00000 0 0) +
    NSP(12 999099NGP00000 00000 0 0) +
     DP(94 99909000000000 00000 0 0)


    SAA(10 004000NGP00113 00000 0 0) =          ( 64. 4)
     U-(90 00300000000000 00000 0 0) +
     XP(11 99900000010000 00000 0 0) +
    NSP(12 999099NGP00000 00000 0 0) +
     DP(94 99909000000000 00000 0 0)


    SAA(10 004010NGP00113 00000 0 0) =          ( 64. 5)
     L-(10 00120000000000 00000 0 0) +
     XP(11 99900000010000 00000 0 0) +
    NSP(12 999099NGP00000 00000 0 0) +
     DP(94 99909000000000 00000 0 0)


    SAB(10 0010YONGP00VIT 00000 0 0) =          ( 65. 1)
    NSP(12 999099NGP00000 00000 0 0) +
     VP(11 9990YONGP99VIT 00000 0 0)


    SAB(10 002000NGP00113 00000 0 0) =          ( 65. 2)
     U-(90 00300000000000 00000 0 0) +
    NSP(10 009009NGP00000 00000 0 0) +
     VP(11 999000NGP01113 00000 0 0)
```

```
      LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z)                 RULE NO.


     SAB(10 003000NGP00113 00000 0 0) =             ( 65. 3)
      U-(10 003000000000000 00000 0 0) +
      R-(10 003000NGP00000 00000 0 0) +
      VP(11 999000NGP99113 00000 0 0)


     SAB(10 004010NGP00113 00000 0 0) =             ( 65. 4)
      L-(10 001200000000000 00000 0 0) +
      R-(10 003000NGP00000 00000 0 0) +
      VP(11 999000NGP99113 00000 0 0)


     SAB(10 005010NGP00113 00000 0 0) =             ( 65. 5)
      L-(10 001200000000000 00000 0 0) +
      NP(10 001009NGP00000 00000 0 0) +
      VP(11 999000NGP99113 00000 0 0)
     SAB(10 006000NGP00113 00000 0 0) +
      NP(10 001009NGP00000 00000 0 0) +
      U-(10 003000000000000 00000 0 0) +
      R-(10 003000NGP00000 00000 0 0) +
      VP(11 999000NGP99113 00000 0 0)


     SAB(10 006010NGP00113 00000 0 0) =             ( 65. 6)
      NP(10 001009NGP00000 00000 0 0) +
      L-(10 001200000000000 00000 0 0) +
      R-(10 003000NGP00000 00000 0 0) +
      VP(11 999000NGP99113 00000 0 0)


     SAC(10 0010YONGP00VIT 00000 317) =             ( 66. 1)
      VP(10 0010YONGP99VIT 00000 317)


      SA(10 0010YONGP00VIT 00000 0 0) =             ( 67. 1)
     SAA(10 0090YONGP00VIT 00000 0 0)
```

LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

```
SYMBOL(MF/KBCLYONGPRAVIT/S/-W-/X/Z)           RULE NO.


   SA(10 0020YONGP00VIT 00000 0 0) =             ( 67. 2)
  SAC(10 0010YONGP00VIT 00000 0 0)


   SA(10 0030YONGP00VIT 00000 0 0) =             ( 67. 3)
  SAB(10 0090YONGP00VIT 00000 0 0)


  SRO(10 0010YONGP00000 00000 0 0) =             ( 68. 1)
  VRB(11 9910YONGP99999 00000 0 0)


  SRO(10 0020YONGP00000 00000 0 0) =             ( 68. 2)
  NSP(12 999099NGP00000 00000 0 0) +
  VRB(11 9910YONGP99999 00000 0 0)


  SRI(10 0010YONGPR0000 00000 0 0) =             ( 69. 1)
  VRI(11 9910YONGPR9999 00000 0 0)


  SRI(10 0020YONGPR0000 00000 0 0) =             ( 69. 2)
  NSP(12 999099NGP00000 00000 0 0) +
  VRI(11 9910YONGPR9999 00000 0 0)


   RG(10 0010YONGP00000 00000 0 0) =             ( 70. 1)
   R-(10 0041Y000000000 00000 0 0) +
   VP(11 999090NGP00999 00000 0 0)


   RG(10 0020YONGP00000 00000 0 0) =             ( 70. 2)
   R-(10 0041Y000000000 00000 0 0) +
  SRO(10 009090999000000 00000 0 0)
```

LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

```
SYMBOL(MF/KBCLYDNGPRAVIT/S/-w-/X/Z)           RULE NO.


   RG(10 0030YDNGPR0000 00000 0 0) =          ( 70. 3)
   R-(10 0041Y000000000 00000 0 0) +
  SR1(10 00909099900000 00000 0 0)


   KN(10 0010Y000000000 00000 0 0) =          ( 71. 1)
   C-(10 00300000000000 00000 0 0) +
   SA(11 9990Y099900999 00000 0 0)


   KN(10 0020Y000000000 00000 0 0) =          ( 71. 2)
   R-(10 00400000000000 00000 0 0) +
   SA(11 9990Y099900999 00000 0 0)


   KC(10 0010Y000000000 00000 0 0) =          ( 72. 1)
   C-(10 00400000000000 99999 0 0) +
   SA(11 9990Y099900999 00000 0 0)


   KC(10 0020Y000000000 00000 0 0) =          ( 72. 2)
   C-(10 00500000000000 99999 0 0) +
   SA(11 9990Y099900999 00000 8 7)


   KC(10 0030Y000000000 00000 0 0) =          ( 72. 3)
   C-(10 00600000000000 99999 0 0) +
   SA(11 9990Y099900999 00000 0 0)


   KC(10 0040Y000000000 00000 0 0) =          ( 72. 4)
   XP(11 9990Y000090000 00000 0 0)
```

```
       LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYONGPRAVIT/S/-W-/X/Z)                  RULE NO.


      KK(10 00100000000001 00000 0 0) =              ( 73. 1)
      U-(10 00400000000000 00000 0 0) +
      SA(11 99900099900911 00000 0 0)


      KK(10 00101000000001 00000 0 0) =              ( 73. 2)
      U-(10 00500000000000 00000 0 0) +
      SA(11 99900099900911 00000 0 0)


      KK(10 0010Y000000002 00000 0 0) =              ( 73. 3)
      U-(10 00600000000000 00000 0 0) +
      SA(11 9990Y099900912 00000 0 0)


      KI(10 0010YONGP00000 00000 0 0) =              ( 74. 1)
      Q-(10 00900000000000 00000 0 0) +
      SA(11 9990YONGP00999 00000 0 0)


      KI(10 0020YONGP00000 00000 0 0) =              ( 74. 2)
      K-(10 00500000000000 99999 0 0) +
      VP(11 9990YONGP09999 00000 0 0)


      KI(10 0030YONGPR0000 00000 0 0) =              ( 74. 3)
      K-(10 00500000000000 99999 0 0) +
     SGO(10 0090YONGPR0000 00000 0 0)


      KI(10 0040YONGPR0000 00000 0 0) =              ( 74. 4)
      P-(10 001000000R0000 00000 0 0) +
      K-(10 005000000R0000 99999 0 0) +
     SRI(10 0090YONGPR0000 00000 0 0)
```

LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

```
SYMBOL(MF/KBCLYDNGFRAVIT/S/-w-/X/Z)          RULE NO.


     KD(10 0010Y00000000 00000 0 0) =          ( 75. 1)
     KN(10 0090Y00000000 00000 0 0)


     KD(10 0020Y00000000 00000 0 0) =          ( 75. 2)
     T-(10 0040000000000 00000 0 0) +
     T-(10 0010000000000 00000 0 0) +
     SC(11 9990Y00000000 00000 0 0) +
     T-(10 0020000000000 00000 0 0)


     SD(10 0010Y000000IT 00000 0 0) =          ( 76. 1)
     KC(11 9990900000000 00000 0 0) +
     T-(90 0030000000000 00000 0 0) +
     SA(11 9990Y0999009IT 00000 0 0)


     SD(10 0020Y000000IT 00000 0 0) =          ( 76. 2)
     SA(11 9990Y099900 9IT 00000 0 0) +
     T-(90 0030000000000 00000 0 0) +
     KC(11 9990900000000 00000 0 0)


     S-(10 0010Y000000IT 00000 0 0) =          ( 77. 1)
     SA(11 9990Y099900 9IT 00000 0 0)


     S-(10 0020Y000000IT 00000 0 0) =          ( 77. 2)
     SD(10 0090Y000000IT 00000 0 0)


     S-(10 0030Y000000I1 00000 0 0) =          ( 77. 3)
     KK(11 9910900000001 00000 0 0) +
     T-(90 0030000000000 00000 0 0) +
     SA(11 9990Y099900 9I1 00000 0 0)
```

```
LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYONSPRAVIT/S/-W-/X/Z)               RULE NO.


      S-(10 0030Y000000012 00000 0 0) =          ( 77. 4)
      KK(11 99109000000002 00000 0 0) +
      T-(90 00300000000000 00000 0 0) +
      SA(11 9990Y099990912 00000 0 0)


      SI(10 0010Y000000000 00000 0 0) =          ( 78. 1)
      KI(10 0090Y099990000 00000 0 0)


      SI(10 0020Y000000000 00000 0 0) =          ( 78. 2)
      KC(11 99909000000000 00000 0 0) +
      T-(10 00300000000000 00000 0 0) +
      KI(10 0090Y099990000 00000 0 0)


      SI(10 0030Y000000000 00000 0 0) =          ( 78. 3)
      KI(10 0090Y099990000 00000 0 0) +
      T-(10 00300000000000 00000 0 0) +
      KC(11 99909000000000 00000 0 0)


      SC(10 0010Y000000000 00000 0 0) =          ( 79. 1)
      C-(90 00800000000000 99999 0 0) +
      S-(10 0090Y000000099 00000 0 0) +
      T-(10 00000000000000 00000 0 0)


      SC(10 0020Y000000000 00000 0 0) =          ( 79. 2)
      C-(90 00800000000000 99999 0 0) +
      SI(11 9990Y000000000 00000 0 0) +
      T-(10 00500000000000 00000 0 0)


      SC(10 0030Y000000000 00000 0 0) =          ( 79. 3)
      C-(90 00800000000000 99999 0 0) +
      S-(10 0090Y000000022 00000 0 0) +
      T-(10 00700000000000 00000 0 0)
```

```
     LISTING OF THE GRAMMAR RUL S OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z)          RULE NO.


      A-(10 001000NG000000 SWWWW 2 9) =            ( 80. 1)
      A-(10 001000NG000000 SWWWW 0 0)


      A-(10 001002NG000000 SWWWW 0 0) =            ( 80. 2)
      B-(10 0010000C000000 00000 0 0) +
      A-(10 001000NG000000 SWWWW 0 0)


      B-(10 0010001G000000 00000 0 0) =            ( 81. 1)
      B-(10 0010001G000000 99999 0 0)


      B-(10 0020002G000000 00000 0 0) =            ( 81. 2)
      B-(10 0020002G000000 99999 0 0)


      B-(10 004000NG000000 00000 0 0) =            ( 81. 3)
      B-(10 004000NG000000 1OSR* 0 0)


      B-(10 005000NG000000 00000 210) =            ( 81. 4)
      B-(10 005000NG000000 1MA** 210)


      B-(10 006000N1000000 00000 0 0) =            ( 81. 5)
      B-(10 006000N1000000 1ALP* 0 0)


      C-(10 00100000000000 00000 0 0) =            ( 82. 1)
      C-(10 00100000000000 1W*** 0 0)
```

```
        LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z)                  RULE NO.


    C-(10 00200000000000 00000 0 0) =               ( 82. 2)
    C-(10 00200000000000 1AW** 0 0)


    C-(10 00300000000000 00000 0 0) =               ( 82. 3)
    C-(10 00300000000000 1KY** 0 0)


    D-(10 00700000000000 00000 0 0) =               ( 83. 1)
    D-(10 00700000000000 1MS** 0 0)


    D-(10 00800000000000 00000 0 0) =               ( 83. 2)
    D-(10 00800000000000 1LKS* 0 0)


    H-(10 00100000000000 00000 0 0) =               ( 84. 1)
    H-(10 00100000000000 1H*** 0 0)


    I-(10 0010001G000000 00000 0 0) =               ( 85. 1)
    I-(10 0010001G000000 99999 0 0)


    I-(10 0020002G000000 00000 0 0) =               ( 85. 2)
    I-(10 0020002G000000 99999 0 0)


    I-(10 004000NG000000 00000 0 0) =               ( 85. 3)
    I-(10 004000NG000000 1OSR* 0 0)


    I-(10 005000NG000000 00000 210) =               ( 85. 4)
    I-(10 005000NG000000 1MA** 210)
```

```
      LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYDNGPRAVIT/S/-w-/X/Z)             RULE NO.


     I-(10 006000N1000000 00000 0 0) =            ( 85. 5)
     I-(10 006000N1000000 1ALP* 0 0)


     L-(10 00110000000000 00000 0 0) =            ( 86. 1)
     L-(10 00110000000000 1LA** 0 0)


     L-(10 00120000000000 00000 0 0) =            ( 86. 2)
     L-(10 00120000000000 1AYN* 0 0)


     L-(10 00130000000000190000 0 0) =            ( 86. 3)
     L-(10 0013000000000019LBLT 0 0)


     L-(10 00140000000000 00000 0 0) =            ( 86. 4)
     L-(10 00140000000000 1AL** 0 0)


     O-(10 00100000000000270000 0 0) =            ( 87. 1)
     O-(10 0010000000000027*AWT 0 0)


     P-(10 00100000010000 00000 0 0) =            ( 88. 1)
     P-(10 00100000010000 1L*** 0 0)


     P-(10 00100000020000 00000 0 0) =            ( 88. 2)
     P-(10 00100000020000 1B*** 0 0)


     P-(10 00100000030000380000 0 0) =            ( 88. 3)
     P-(10 0010000003000038KMw* 0 0)
```

```
LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYUNGPRAVIT/S/-W-/X/Z)                RULE NO.


P-(10 0010000004000 00000 0 0) =                   ( 88. 4)
P-(10 0010000004000 1M*** 0 0)


P-(10 001000000500038000 0 0) =                    ( 88. 5)
P-(10 0010000005000038MYMN 0 0)


P-(10 00100000060000270000 0 0) =                  ( 88. 6)
P-(10 001000000600027ALY* 0 0)


P-(10 00100000070000270000 0 0) =                  ( 88. 7)
P-(10 0010000007000270LY* 0 0)


P-(10 0010000008000 00000 0 0) =                   ( 88. 8)
P-(10 0020000008000 99999 0 0)


G-(10 0010000000000 00000 0 0) =                   ( 90. 1)
G-(10 0010000000000 1H*** 0 0)


G-(10 0020000000000 00000 0 0) =                   ( 90. 2)
G-(10 0020000000000 1HAM* 0 0)


G-(10 0030000000000 00000 0 0) =                   ( 90. 3)
G-(10 0030000000000 99999 0 0)


T-(10 0010000000000 00000 0 0) =                   ( 91. 1)
T-(10 0010000000000 1''** 0 0)
```

```
          LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

    SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z)              RULE NO.


       T-(10 0020000000000000 00000 0 0) =             ( 91. 2)
       T-(10 0020000000000000 1''** 0 0)


       T-(10 0030000000000000 00000 0 0) =             ( 91. 3)
       T-(10 0030000000000000 1,*** 0 0)


       T-(10 0040000000000000 00000 0 0) =             ( 91. 4)
       T-(10 0040000000000000 1:*** 0 0)


       T-(10 0050000000000000 00000 0 0) =             ( 91. 5)
       T-(10 0050000000000000 1?*** 0 0)


       T-(10 0060000000000000 00000 0 0) =             ( 91. 6)
       T-(10 0060000000000000 1.*** 0 0)


       T-(10 0070000000000000 00000 0 0) =             ( 91. 7)
       T-(10 0070000000000000 1!*** 0 0)


       T-(10 0080000000000000 00000 0 0) =             ( 91. 8)
       T-(10 0080000000000000 1-*** 0 0)


       U-(10 0010000000000000 00000 0 0) =             ( 92. 1)
       U-(10 0010000000000000 1SL** 0 0)


       U-(10 0020000000000000 00000 0 0) =             ( 92. 2)
       U-(10 0020000000000000 1NA** 0 0)
```

```
   LISTING OF THE GRAMMAR RULES OF MODERN HEBREW SYNTAX

SYMBOL(MF/KBCLYDNGPRAVIT/S/-W-/X/Z)              RULE NO.


    U-(10 0030000000000 00000 0 0) =              ( 92. 3)
    U-(10 0030000000000 1YS** 0 0)


    U-(10 0040000000000 00000 0 0) =              ( 92. 4)
    U-(10 0040000000000 1LW** 0 0)


    U-(10 0050000000000 00000 0 0) =              ( 92. 5)
    U-(10 0050000000000 1LWLA 0 0)


    U-(10 0060000000000 00000 0 0) =              ( 92. 6)
    U-(10 0060000000000 1AM** 0 0)


    U-(10 0070000000000 00000 0 0) =              ( 92. 7)
    U-(10 0070000000000 1H*** 0 0)


    V-(10 0010YONGP01VIT12HYH* 0 0) =              ( 93. 1)
    V-(10 0010YONGP01VIT12HYH* 0 0)


    V-(10 001000NGPRAVIT SWWWW 0 0) =              ( 93. 2)
    V-(10 001000NGPRAVIT SWWWW 0 0)


    E-(10 001000NGPRAVIT SWWWW 0 0) =              ( 94. 1)
    E-(10 001000NGPRAVIT SWWWW 0 0)
```

Appendix

PART III

APPENDIX B

SOURCE LANGUAGE LISTING
OF
COMPUTER PROGRAM SENSYN
AND
ASSOCIATED SUBPROGRAMS

PART III

APPENDIX B

This appendix contains the source language listing of the following computer programs and associated subprograms:

(1)	Main Program	SENSYN
(2)	Subprogram	ALPHA
(3)	"	ALPHA1
(4)	"	CONSEN
(5)	"	DIAGRM
(6)	"	INPUT
(7)	"	LIMIT
(8)	"	ORTHOG
(9)	"	OUTPUT
(10)	"	RERITE
(11)	"	RULENO
(12)	"	RULRIT
(13)	"	SYMACH
(14)	"	VARATT
(15)	"	INFLEX

MAIN PROGRAM

SENSYN

```
C
C     PROGRAM SENSYN-SYNTHESIZES SENTENCES IN HEBREW
C
      COMMON  RULE(800,28),IPSI(100,8,2),RESTRT(5,15),ITABLE(2,50,29),
     1 ILIST,ILIST1,I,J,ITERM,SYMIN(29),IA,IRULE,IKLASS,SYMBOL(29),IB,
     2 IMAX,IEND,VARSTR(29),IVAR(10,29),SYMSTR(10,29),IREG(10,3),MATCH,
     3 ITRACE
      COMMON/ABC/ TRANSL(50)
      COMMON/TREE/ ITREE(50),NODE(5,400), NODE1
      COMMON/SYMB/ SYM(100)
      INTEGER RULE, RESTRT,SYMIN,SYMBOL, VARSTR,SYMSTR
      LOGICAL MATCH
      DIMENSION REEDER(36),ROOT(4),ENGLSH(13)
      READ(5,102) MODE,ITAPE, ITRACE, IOUTPT,IOUTRE
      REWIND ITAPE
      IF(MODE.EQ.0) GO TO 10
      READ(ITAPE) TRANSL,RULE,IPSI,RESTRT ,SYM
      REWIND ITAPE
      GO TO 4
C
C ***                   READ IN TRANSLITERATION
C
   10 READ(5,100) NL,(TRANSL(L),L=1,NL)
  100 FORMAT(I6,50A1)
C
C ***                   READ IN GRAMMAR RULES
C
      I=0
      ISYMNO=0
      ICLASS=0
      IRULE=0
    1 I=I+1
      READ(5,101) (REEDER(L),L=1,36)
  101 FORMAT(3A1,2(2X,A1),1X,2A1,13(2X,A1),1X,2A1,5(2X,A1),9A1)
      RULE(I,1)=0
      DO 11 L=1,3
      CALL ALPHA(REEDER(L),N)
   11 RULE(I,1)=RULE(I,1)+ N*10**(3-L)
      IF(RULF(I,1).EQ.999) GO TO 3
      DO 12 L=4,5
      L1=L-2
      CALL ALPHA(REEDER(L), N)
   12 RULE(I,L1)=N
      CALL ALPHA(REEDER(6),II)
      CALL ALPHA(REEDER(7),N)
      RULE(I,4)=N
      IF(II.EQ.39) RULE(I,4)=-RULE(I,4)
      DO 121 L=8,20
      L1=L-3
      CALL ALPHA(REEDER(L),N)
  121 RULE(I,L1)=N
      CALL ALPHA(REEDER(21),N)
      CALL ALPHA(REEDER(22),N1)
      RULE(I,18)=N1+N*10
      IF(N.GT.0)  RULE(I,18)=-RULE(I,18)
      DO 122 L=23,27
      L1=L-4
      CALL ALPHA(REEDER(L),N)
  122 RULE(I,L1)=N
      RULE(I,24)=0
      DO 13 L=28,30
      CALL ALPHA(REEDER(L),N)
   13 RULE(I,24)=RULE(I,24)+ N*10**(30-L)
      DO 14 L=31,32
      L1=L-6
      CALL ALPHA(REEDER(L),N)
   14 RULE(I,L1)=N
      RULE(I,27)=0
      DO 15 L=33,34
      CALL ALPHA(REEDER(L),N)
   15 RULE(I,27)=RULE(I,27)+ N*10**(34-L)
      RULE(I,28)=0
      DO 16 L=35,36
      CALL ALPHA(REEDER(L),N)
   16 RULE(I,28)=RULE(I,28)+N*10**(36-L)
      IF(RULE(I,25).NE.0) GO TO 2
      IF(RULE(I,6).EQ.ICLASS.AND.RULE(I,24).EQ.IRULE) GO TO 2
      ISYMNO=RULE(I,1)
      ICLASS=RULE(I,6)
      IRULE=RULE(I,24)
      IF(ISYMNO.GT.96)  GO TO 1
      IPSI(ISYMNO,ICLASS,1)=I
      GO TO 1
    2 IPSI(ISYMNO,ICLASS,2)=I
      GO TO 1
C
C ***                    READ IN RESTRICTION MATRIX
C
    3 IRULE=I
      READ(5,102)((RESTRT(I,J),I=1,5),J=1,15)
  102 FORMAT(5I5)
C
C ***                    READ IN SYMBOL ARRAY
C
      READ(5,104) SYM
  104 FORMAT(10(2X,A3))
      WRITE(ITAPE) TRANSL,RULE,IPSI,RESTRT, SYM
      REWIND ITAPE
      WRITE(6,200) TRANSL
  200 FORMAT(5X,50A1)
      WRITE(6,201)(NN,(RULE(NN,L),L=1,28),NN=1,IRULE)
  201 FORMAT(29I4)
      WRITE(6,202)(MM,((IPSI(MM,NN,L),L=1,2),NN=1,8),MM=1,100)
  202 FORMAT(17I5)
      WRITE(6,203)((RESTRT(NN,L),NN=1,5),L=1,15)
  203 FORMAT(5I5)
```

```
      WRITE(6,104) SYM
C
C ***                  CLEAR REWRITE LIST MATRI
C
    4 DO 5 I=1,2
      DO 5 J=1,50
      ITREE(J)=0
      DO 5 K=1,29
    5 ITABLE(I,J,K)=0
      DO 51 I=1,5
      DO 51 J=1,300
   51 NODE(I,J)=0
      NODE1=0
C
C****                  READ ENGLISH SENTENCE AND INITIAL SYMBOL
C
      READ(5,105) ENGLSH
  105 FORMAT(13A6)
      READ(5,103) (ITABLE(1,1,K),K=1,18),(ROOT(L),L=1,4)
  103 FORMAT(I6,4X,17I2,4A1)
      IF(ITABLE(1,1,1).EQ.999) GO TO 999
      ITABLE(1,1,26)=1
      DO 6 I=1,4
      I1=I+18
      CALL ALPHA(ROOT(I),NN)
    6 ITABLE(1,1,I1)=NN
      ILIST=0
      IPASS=0
      J=1
      ILIST1=1
      IF(IOUTPT.EQ.1) CALL OUTPUT(IPASS)
    7 ILIST=ILIST+1
      ILIST1=2
      IMAXI=J
      IPASS=IPASS+1
      ITREE(IPASS)=IMAXI
      J=0
      ITERM=0
      IF(ILIST.GT.2) ILIST=1
      IF(ILIST.EQ.2) ILIST1=1
      DO 8 I=1,IMAXI
      ICONC=ITABLE(ILIST,I,3)
      ICOMP=ITABLE(ILIST,I,4)
      ICONB=ITABLE(ILIST,I,5)
      ICOMP1=ICOMP-1
      IF(ICONC.EQ.3.OR.ICONB.EQ.3) ICOMP1=ICOMP
      IF(ICONC.EQ.4) ICOMP1=0
      CALL RERITE
      NODE1=NODE1+1
      NODE(1,NODE1)=ITABLE(ILIST,I,26)
      IF(ICOMP.NE.0)  NODE(1,NODE1)=ICOMP+ ICOMP1
      NODE(4,NODE1)=ITABLE(ILIST,I,1)
    8 NODE(5,NODE1)=ITABLE(ILIST,I,6)
      IF(IOUTPT.EQ.1) CALL OUTPUT(IPASS)
      IF(ITERM.LT.J) GO TO 7
      IPASS=IPASS+1
      ITREE(IPASS)=ITERM
      DO 9 I=1,ITERM
      NODE1=NODE1+1
      NODE(1,NODE1)=ITABLE(ILIST1,I,26)
      NODE(4,NODE1)=ITABLE(ILIST1,I,1)
    9 NODE(5,NODE1)=ITABLE(ILIST1,I,6)
      IPASS=IPASS+1
      ITREE(IPASS)=ITERM
      CALL CONSEN
      WRITE(6,106) ENGLSH
  106 FORMAT(1H1//10X,28HEQUIVALENT ENGLISH SENTENCE-/10X,13A6)
      IF(IOUTRE.EQ.1) CALL DIAGRM(IPASS)
      CALL ORTHOG
      GO TO 4
  999 WRITE(6,199)
  199 FORMAT(1H1,10(/),1X,17(7H--END--))
      STOP
      END
```

Subprogram ALPHA

```
      SUBROUTINE ALPHA(A,IA)
C
C *** THIS SUBROUTINE CONVERTS THE ALPHA-NUMERIC SYMBOL(A)  TO AN INTEGER
C *** EQUIVALENT BASED ON DATA STORED IN ARRAY TRANSL
C
      COMMON/ABC/ TRANSL(50)
C
      DO 1 L=1,50
      IF(A-TRANSL(L)) 1,2,1
    1 CONTINUE
      IA=49
      RETURN
    2 IA=L-1
      IF(IA.EQ.40) IA=0
      RETURN
      END
```

Subprogram ALPHA1

```
      SUBROUTINE ALPHA1(A,IA)
C
C *** THIS SUBROUTINE CONVERTS THE ALPHA-NUMERIC SYMBOL(A)  TO AN INTEGER
C *** EQUIVALENT BASED ON DATA STORED IN ARRAY TRANSL
C                 AND CONVERTS SPACE TO *
C
      COMMON/ABC/ TRANSL(50)
C
      DO 1 L=1,50
      IF(A-TRANSL(L)) 1,2,1
    1 CONTINUE
      IA=49
      RETURN
    2 IA=L-1
      IF(IA.EQ.40) IA=44
      RETURN
      END
```

Subprogram CONSEN

```
      SUBROUTINE CONSEN
C
C ***     THIS SUBROUTINE REWRITES THE FINAL STRING OF SYMBOLS AND PERFORM
C ***     CONTEXT SENSITIVE TRANSFORMS ON APPLICABLE SYMBOLS
C
      COMMON  RULE(800,28),IPSI(100,8,2),RESTRT(5,15),ITABLE(2,50,29),
     1 ILIST,ILIST1,I,J,ITERM,SYMIN(29),IA,IRULE,IKLASS,SYMBOL(29),IB,
     2 IMAX,IEND,VARSTR(29),IVAR(10,29),SYMSTR(10,29),IREG(10,3),MATCH,
     3 ITRACE
      COMMON/TREE/ ITREE(50),NODE(5,400), NODE1
      INTEGER   R1,R2,R3,R4
      EQUIVALENCE (IS,J)
      IF(ITRACE.NE.0) WRITE(6,99)
   99 FORMAT(    30H     SUBROUTINE CONSEN CALLED )
      I=ILIST
      ILIST=ILIST1
      ILIST1=I
      IT=ITERM
      IS=0
      DO 20 I=1,IT
      IJ=1
      II=I-1
      IH=ITABLE(ILIST,I,24)
      ITA=ITABLE(ILIST,I,1)
      ITC=ITABLE(ILIST,I,6)
      IF(ITA.NE.83.AND.(.NOT.(ITA.EQ.91.AND.ITC.EQ.3))) GO TO 18
      ITA=ITABLE(ILIST,I1,1)
      ITC=ITABLE(ILIST,I1,6)
      IN =ITABLE(ILIST,I1,10)
      IG =ITABLE(ILIST,I1,11)
      IP =ITABLE(ILIST,I1,12)
      IR =ITABLE(ILIST,I1,13)
      IM =ITABLE(ILIST,I1,16)
      ITT=ITABLE(ILIST,I1,17)
       IF(ITABLE(ILIST,I,1).EQ.91) GO TO 1
       IF(ITA.EQ.89.AND.IR.LT.4) GO TO 10
       GO TO 18
C
C ***      RULES FOR ACCUSATIVE PRONOMINAL SUFFIX
C
     1 ITA=ITA-76
       GO TO(9,9,9,9,9,2,9,2,2,3,9,4,5,9,9,9,6,7,9,8,9,5), ITA
C
C            7 7 7 8 8 8 8 8 8 8 8 8 8 9 9 9 9 9 9 9 9 9
C            7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8
```

```
C
C ***                  RULES FOR PRONOUN FOLLOWING PARTICIPLE, NUMBER, OR NOUN
C
    2 IJ=3
      IF(IN.GT.1.AND.IG.EQ.2) IH=1
      GO TO 9
C
C ***                  RULE FOR PRONOUN FOLLOWING AYYN
C
    3 IF(ITC.EQ.2) IJ=2
      GO TO 9
C
C ***                  RULE FOR PRONOUN FOLLOWING AWT
C
    4 IJ=4
      GO TO 9
C
C                      RULES FOR PRONOUN FOLLOWING PREPOSITIONS
C
    5 GO TO(53,53,55,56,56,53,53,58), IR
C
C---- IR=    1, 2, 3, 4, 5, 6, 7, 8
C
   53 IJ=3
      GO TO 9
   55 IJ=5
      GO TO 9
   56 IJ=6
      ITABLE(ILIST1,IS,13)=5
      GO TO 9
   58 R1= ITABLE(ILIST,I1,19)
      R2=ITABLE(ILIST,I1,20)
      R3=ITABLE(ILIST,I1,21)
      R4=ITABLE(ILIST,I1,22)
      IF(R1.EQ.10.AND.R2.EQ.27.AND.R3.EQ.21.AND.R4.EQ.44) GO TO 4
      IJ=3
      GO TO 9
C
C ***                  RULES FOR PRONOUN FOLLOWING  YS,  AND SL
C
    6 IF(ITC.EQ.3) IJ=6
      IF(ITC.EQ.1) IJ=3
      GO TO 9
```

```
C
C ***            RULES FOR PRONOUN FOLLOWING VERBS
C
    7 IJ=2
      IF(IT.NE.2) GO TO 9
      IF(IP.EQ.1.AND.IM.EQ.1) GO TO 71
      IF(IN.EQ.1.AND.IG.EQ.1.AND.IP.EQ.2) GO TO 71
      IF(IN.EQ.1.AND.IP.EQ.3.AND.IM.EQ.1) GO TO 71
      GO TO 9
   71 IH=1
      GO TO 9
C
C ***            RULES FOR PRONOUN FOLLOWING INFINITIVE
C
    8 IJ=3
      IH=1
      GO TO 9
C
C ***            REWRITE SYMBOL
    9 ITABLE(ILIST1,IS,23)=3
   18 ITABLE(ILIST,I,23)=IJ
      ITABLE(ILIST,I,24)=IH
      IS=IS+1
      NODE1=NODE1+1
      NODE(1,NODE1)=1
      NODE(4,NODE1)=ITABLE(ILIST,I,1)
      NODE(5,NODE1)=ITABLE(ILIST,I,6)
      DO 19 L=1,29
   19 ITABLE(ILIST1,IS,L)=ITABLE(ILIST,I,L)
      GO TO 20
C
C ***       RULE FOR DEF. ART. FOLLOWING PREP. B,C,L
C
   10 ITERM=ITERM-1
      NODE1=NODE1+1
      NODE(1,NODE1)=0
      NODE(4,NODE1)=ITABLE(ILIST,I,1)
      NODE(5,NODE1)=ITABLE(ILIST,I,6)
   20 CONTINUE
      RETURN
      END
```

Subprogram DIAGRM

```
      SUBROUTINE DIAGRM(IPASS)
C
C ***      THTS SUBROUTINE CONSTRUCTS A TREE DIAGRAM OF THE GENERATED SEN
C
      COMMON  RULE(800,28),IPSI(100,8,2),RESTRT(5,15),ITABLE(2,50,29),
     1 ILIST,ILIST1,I,J,ITERM,SYMIN(29),IA,IRULE,IKLASS,SYMBOL(29),IB,
     2 IMAX,IEND,VARSTR(29),IVAR(10,29),SYMSTR(10,29),IREG(10,3),MATCH,
     3 ITRACE
      COMMON/TREE/ ITREE(50),NODE(5,400), NODE1
      COMMON/SYMB/ SYM(100)
      DIMENSION ALINE(40), CLS(9)
      DATA BLANK/3H   /
      DATA DASHES/3H---/
      DATA BAR/3H  I/
      DATA CLS/3H1  ,3H2  ,3H3  ,3H4  ,3H5  ,3H6  ,3H7  ,3H8  ,3H9  /
      IF(ITRACE.NE.0) WRITE(6,99)
   99 FORMAT(30H     SUBROUTINE DIAGRAM CALLED  )
C
C ***                    COMPUTE GOVERNING NODE OF EACH NODE
C
      L=0
      M=0
      L1=1
    1 L=L+1
      N=NODE(1,L)
      IF(N.GT.0.OR.(N.EQ.0.AND.L.GT.NODE1)) 60 TO 91
      NG=NODE(3,L)
   90 CONTINUE
      NODE(1,N6)=NODE(1,NG)-1
      IF(NODE(1,NG).NE.0) GO TO 1
      NG=NODE(3,NG)
      GO TO 90
   91 DO 2  I=1,N
      L1=L1+1
    2 NODE(3,L1)=L
      IF(L1.LT.NODE1) GO TO 1
C
C ***                    COMPUTE LINE POSITION FOR EACH NODE
C
      NP=ITREE(IPASS)
      NP1=NODE1-NP+1
      DO 3  L=NODE1,NP1,-1
      NODE(2,L)=NP
    3 NP=NP-1
      L=NODE1+1
    4 L=L-1
      IF(NG.GT.0.AND.NODE(3,L).EQ.NG) GO TO 4
      IF(NODE(1,L).EQ.0.AND.NODE(2,L).EQ.0) GO TO 6
      NG=NODE(3,L)
      IF(NODE(1,NG).GT.1) GO TO 5
      NODE(2,NG)=NODE(2,L)
      GO TO 6
    5 L1=L+1-NODE(1,NG)
      NODE(2,NG)= (NODE(2,L)+NODE(2,L1))/2
      L=L-NODE(1,NG)+1
    6 IF(L.GT.1) GO TO 4
      IF(ITRACE.EQ.0) GO TO 71
      WRITE(6,103)
  103 FORMAT(1H1//10X,11HNODE MATRIX//)
      DO 7 L=1,NODE1
    7 WRITE(6,102)(NODE(I,L),I=1,5)
  102 FORMAT(10X,5I5)
   71 CONTINUE
C
C ***                  WRITE OUT TREE DIAGRAM
C
      WRITE(6,100)
  100 FORMAT(//10X,35H TREE DIAGRAM OF HEBREW SENTENCE   //)
      NO=1
      DO 50 M=1,IPASS
      IF(M.EQ.1) GO TO 31
C
C ***                  WRITE UPPER CONNECTORS (HORIZONTAL)
C
      IEND=0
      IENDPO=0
      DO 20 L=1,20
      IF((NODE(1,NO).EQ.0.OR.NODE(2,NO).EQ.0).AND.NO.LE.NODE1) NO=NO+1
      L1=L+L-1
      L2=L+L
      IF(L.EQ.IENDPO)  IEND=1
      IF(L-NODE(2,NO))  10,11,12
   10 ALINE(L1)=BLANK
      ALINE(L2)=BLANK
      GO TO 20
   11 NG=NODE(3,NO)
      NO1=NODE(1,NG)
      NO2=NO+NO1-1
      IENDPO=NODE(2,NO2)
      ALINE(L1)=BLANK
      IF(IENDPO.GT.L) GO TO 13
      ALINE(L1)=BAR
      NO=NO+1
      GO TO 20
   12 IF(IEND.EQ.1) GO TO 14
      IF(L.GT.IENDPO) GO TO 10
      ALINE(L1)=DASHES
   13 ALINE(L2)=DASHES
      GO TO 20
   14 ALINE(L1)=DASHES
      ALINE(L2)=BLANK
      NO=NO+NO1
      IEND=0
      IENDPO=0
   20 CONTINUE
      NO=NO-ITREE(M)
      IUPLOW=1
      WRITE(6,101) ALINE
  101 FORMAT(1X,40A3)
C
C ***                  WRITE UPPER CONNECTORS(VERTICAL)
C
   21 DO 30 L=1,20
      IF((NODE(1,NO).EQ.0.OR.NODE(2,NO).EQ.0).AND.NO.LE.NODE1) NO=NO+1
      L1=L+L-1
      L2=L+L
      IF(L.EQ.NODE(2,NO)) GO TO 22
      ALINE(L1)=BLANK
      ALINE(L2)=BLANK
      GO TO 30
   22 ALINE(L2)=BLANK
      ALINE(L1)=BAR
      NO=NO+1
   30 CONTINUE
      NO=NO-ITREE(M)
      WRITE(6,101) ALINE
      IF(IUPLOW.EQ.2) GO TO 50
C
C ***                  WRITE LINE OF NODES
C
   31 DO 40 L=1,20
      IF((NODE(1,NO).EQ.0.OR.NODE(2,NO).EQ.0).AND.NO.LE.NODE1) NO=NO+1
      L1=L+L-1
      L2=L+L
      IF(L.EQ.NODE(2,NO)) GO TO 32
      ALINE(L1)=BLANK
      ALINE(L2)=BLANK
      GO TO 40
   32 NS=NODE(4,NO)
      NT=NODE(5,NO)
      NG=NODE(3,NO)
      NO=NO+1
      IF(NS.EQ.NODE(4,NG).AND.NT.EQ.NODE(5,NG)) GO TO 34
   33 ALINE(L1)=SYM(NS)
      ALINE(L2)=CLS(NT)
      GO TO 40
   34 IF(M.EQ.IPASS) GO TO 33
      ALINE(L2)=BLANK
      ALINE(L1)=BAR
   40 CONTINUE
      NO=NO-ITREE(M)
      WRITE(6,101) ALINE
C
C ***                  WRITE LOWER CONNECTORS (VERTICAL)
C
   41 IUPLOW=2
      IF(M.NE.IPASS) GO TO 21
   50 NO=NO+ ITREE(M)
      RETURN
      END
```

Subprogram INPUT

```
      SUBROUTINE INPUT
C
C *** THIS SUBROUTINE CHECKS THE SUBSCRIPT VARAIBLES OF A GIVEN SYMBOL
C *** AND READS IN THE VALUES FOR ANY INDEPENDENT VARIABLES
C
      COMMON  RULE(800,28),IPSI(100,8,2),RESTRT(5,15),ITABLE(2,50,29),
     1 ILIST,ILIST1,I,J,ITERM,SYMIN(29),IA,IRULE,IKLASS,SYMBOL(29),IB,
     2 IMAX,IEND,VARSTR(29),IVAR(10,29),SYMSTR(10,29),IREG(10,3),MATCH,
     3 ITRACE
      COMMON/INR/  INREAD(29)
      INTEGER RULE,RESTRT,SYMIN,SYMBOL,VARSTR,SYMSTR
      LOGICAL MATCH
      DIMENSION REEDER(4)
      IF(ITRACE.NE.0) WRITE(6,99)
   99 FORMAT(     30H     SUBROUTINE INPUT  CALLED )
      DO 9 L=1,29
    9 INREAD(L)=0
      MATCH=.TRUE.
      N=22
      IF(SYMIN(1).NE.SYMBOL(1).AND.SYMBOL(1).NE.97) GO TO 7
      DO 8 L=2,N
    8 IF(SYMIN(L).EQ.9) SYMBOL(L)=9
    7 DO 4 L=2,N
      IF(SYMBOL(L).NE.9) GO TO 4
      READ(5,100)(INREAD(M),M=1,18),REEDER,(INREAD(M),M=23,29)
  100 FORMAT(I6,4X,17I2,4A1,7I2)
      DO 1 M=1,4
      M1=M+18
      CALL ALPHA(REEDER(M),NN)
    1 INREAD(M1)=NN
      IF(INREAD(1).NE.SYMBOL(1).AND.SYMBOL(1).NE.97) MATCH=.FALSE.
      GO TO 5
    4 CONTINUE
    5 DO 3 M=2,N
      IF(SYMBOL(M).GE.-1.AND.SYMBOL(M).LE.9)  GO  TO 2
      IF(VARSTR(M).EQ.0.AND.M.GT.18.AND.SYMBOL(1).NE.97) GO TO 3
      IF(VARSTR(M).EQ.0) VARSTR(M)=INREAD(M)
      SYMBOL(M)=VARSTR(M)
      GO TO 3
    2 IF(SYMBOL(M).EQ.9.AND.(INREAD(4).EQ.0.OR.M.LE.6) )
     1  SYMBOL(M)=INREAD(M)
      IF(SYMBOL(M).EQ.-1) SYMBOL(M)=VARSTR(M)-1
    3 CONTINUE
      SYMBOL(24)=INREAD(24)
      SYMBOL(29)=INREAD(25)
      IF(SYMBOL(1).EQ.97) SYMBOL(1)=VARSTR(1)
      IF(SYMBOL(27).NE.0.OR.(SYMIN(3).EQ.2.AND.SYMIN(4).GT.0.AND.
     1  SYMBOL(1).LT.77))  CALL LIMIT
      IF(ITRACE.NE.0) WRITE(6,200) SYMBOL
  200 FORMAT(29I4)
      RETURN
      END
```

Subprogram LIMIT

```
      SUBROUTINE LIMIT
C
C ***                   VERSION 2
C *** THIS SUBROUTINE CHECKS A GIVEN SYMBOL AGAINST SPECIFIED LIMITATIO
C *** IF LIMITATIONS ARE NOT SATISFIED,VARIABLE MATCH IS SET TO FALSE.
C
      COMMON  RULE(800,28),IPSI(100,8,2),RESTRT(5,15),ITABLE(2,50,29),
     1 ILIST,ILIST1,I,J,ITERM,SYMIN(29),IA,IRULE,IKLASS,SYMBOL(29),IB,
     2 IMAX,IEND,VARSTR(29),IVAR(10,29),SYMSTR(10,29),IREG(10,3),MATCH,
     3 ITRACE
      INTEGER RULE,RESTRT,SYMIN,SYMBOL,VARSTR,SYMSTR
      LOGICAL MATCH
      IF(ITRACE.NE.0) WRITE(6,99)
   99 FORMAT(    30H     SUBROUTINE LIMIT  CALLED )
      IF(SYMIN(3).EQ.2.AND.SYMIN(4).GT.0) GO TO 4
      IR=SYMBOL(27)
      IV=SYMBOL(28)
      IM=RESTRT(1,IR)+1
      DO 2 L=2,IM
      IF(RESTRT(L,IR).GE.0) GO TO 1
      MATCH=.TRUE.
      IREST=-RESTRT(L,IR)
      IF(SYMBOL(IV).NE.IREST) GO TO 2
      GO TO 3
    1 IF(SYMBOL(IV).EQ.RESTRT(L,IR)) GO TO 14
      MATCH=.FALSE.
    2 CONTINUE
      RETURN
    3 MATCH=.FALSE.
      RETURN
    4 IF(SYMIN(11).NE.2) GO TO 6
    5 IF(SYMBOL(11).NE.2) GO TO 3
    6 IF(SYMIN(12).NE.2) GO TO 8
    7 IF(SYMBOL(12).EQ.1) GO TO 3
    8 IF(SYMIN(12).NE.3) GO TO 10
    9 IF(SYMBOL(12).NE.3) GO TO 3
   10 IF(SYMIN(5).NE.2) GO TO 14
   11 IF(SYMIN(10).NE.1) GO TO 14
   12 IF(SYMBOL(10).NE.1) GO TO 3
   14 MATCH=.TRUE.
      RETURN
      END
```

Subprogram ORTHOG

```
      SUBROUTINE ORTHOG
C
C *** THIS SUBROUTINE ASSEMBLES THE WORDS OF A HEBREW SENTENCE.
C *** IT USES SUBROUTINE INFLEX TO COMPUTE THE CORRECT ORTHOGRAPHY OF EACH
C *** WORD.  IT EDITS A STRING OF SUCH WORDS TO INSERT SPACES WHERE REQUIRED.
C
      COMMON  RULE(800,28),IPSI(100,8,2),RESTRT(5,15),ITABLE(2,50,29),
     1 ILIST,ILIST1,I,J,ITERM,SYMIN(29),IA,IRULE,IKLASS,SYMBOL(29),IB,
     2 IMAX,IEND,VARSTR(29),IVAR(10,29),SYMSTR(10,29),IREG(10,3),MATCH,
     3 ITRACE
      COMMON/ABC/ TRANSL(50)
      DIMENSION ALINE(140),WORD(19)
      INTEGER SYMBOL, WORD
      IF(ITRACE.NE.0) WRITE(6,99)
   99 FORMAT(     30H     SUBROUTINE ORTHOG CALLED )
      ISPACE=0
      WRITE(6,100)
  100 FORMAT(//10X,26HRESULTANT HEBREW SENTENCE-//)
C
C ***                    ERASE PREVIOUS LINE
C
    1 DO 2  L=1,140
    2 ALINE(L)=TRANSL(41)
C
C ***                    COMPUTE ORTHOGRAPHY OF WORDS
C
      DO 13  M=1,ITERM
      DO 3   L=1,29
    3 SYMBOL(L)=ITABLE(ILIST1,M,L)
      CALL INFLEX(SYMBOL,WORD,NC)
C
C ***                    CHECK FOR END OF LINE, IF SO, EDIT AND PRINT
C
      ICHECK=ISPACE+NC-M
      IF(ICHECK.LT.120) GO TO 11
      IF(WORD(1).EQ.47)  ALINE(ISPACE)=TRANSL(40)
   14 IS=0
      DO 9 L=2,140
      L1=L-1
      IF(ALINE(L)-TRANSL(46)) 4,6,4
    4 IF(ALINE(L)-TRANSL(47)) 5,9,5
    5 IS=IS+1
      ALINE(IS)=ALINE(L)
      GO TO 9
    6 IF(ALINE(L1)-TRANSL(46)) 9,8,9
    8 IS=IS+1
      ALINE(IS)=TRANSL(41)
    9 CONTINUE
      I6=IS+1
      DO 91 L=IS,140
   91 ALINE(L)=TRANSL(41)
      WRITE(6,101)(ALINE(L),L=1,120)
  101 FORMAT(    5X,120A1)
      ISPACE=0
      DO 10 L=1,140
   10 ALINE(L)=TRANSL(41)
      IF(M.EQ.ITERM) GO TO 13
   11 DO 12 L=1,NC
      ISPACE=ISPACE+1
      IW=WORD(L)+1
   12 ALINE(ISPACE)=TRANSL(IW)
      IF(M.EQ.ITERM) GO TO 14
   13 CONTINUE
      RETURN
      END
```

Subprogram OUTPUT

```
      SUBROUTINE OUTPUT(ITH)
C
C *** THIS SUBROUTINE WRITES THE RESULTS OF THE ITH REWRITE PASS OF THE
C *** GRAMMAR
C
      COMMON  RULE(800,28),IPSI(100,8,2),RESTRT(5,15),ITABLE(2,50,29),
     1 ILIST,ILIST1,I,J,ITERM,SYMIN(29),IA,IRULE,IKLASS,SYMBOL(29),IB,
     2 IMAX,IEND,VARSTR(29),IVAR(10,29),SYMSTR(10,29),IREG(10,3),MATCH,
     3 ITRACE
      COMMON/ABC/ TRANSL(50)
      INTEGER RULE, RESTRT,SYMIN,SYMBOL,VARSTR,SYMSTR
      LOGICAL  MATCH
      IF(ITRACE.NE.0) WRITE(6,99)
   99 FORMAT(    30H     SUBROUTINE OUTPUT CALLED )
      WRITE(6,100) ITH
  100 FORMAT(1H1///9X,28HRESULTS OF REWRITE PASS NO.-,I3//
     1105H     SYMBOL    M  F  K  B  C  L  Y  D  N  G  P  R  A  V  I  T
     2 6     W       TY  RULE EL NE  REST SUB    X//)
      DO 1 JJ=1,J
      IR1=ITABLE(ILIST1,JJ,19) +1
      IR2=ITABLE(ILIST1,JJ,20) +1
      IR3=ITABLE(ILIST1,JJ,21) +1
      IR4=ITABLE(ILIST1,JJ,22) +1
    1 WRITE(6,101) (ITABLE(ILIST1,JJ,L),L=1,18),TRANSL(IR1),
     1 TRANSL(IR2), TRANSL(IR3), TRANSL(IR4), (ITABLE(ILIST1,JJ,L),
     2 L=23,29)
  101 FORMAT(6X,I3,4X,17I3,4X,4A1,3X,I3,3X,3I3,3(2X,I3))
      RETURN
      END
```

Subprogram RERITE

```
      SUBROUTINE RERITE
C
C ***                VERSION 2
C ***  THIS SUBROUTINE REWRITES STRINGS OF SYMBOLS USING SUBROUTINE RULENO
C ***   TO COMPUTE THE APPLICABLE RULE NUMBER AND SUBROUTINE RULRIT TO
C ***   APPLY THE APPROPRIATE RULE
C
      COMMON  RULE(800,28),IPSI(100,8,2),RESTRT(5,15),ITABLE(2,50,29),
     1 ILIST,ILIST1,I,J,ITERM,SYMIN(29),IA,IRULE,IKLASS,SYMBOL(29),IB,
     2 IMAX,IEND,VARSTR(29),IVAR(10,29),SYMSTR(10,29),IREG(10,3),MATCH,
     3 ITRACE
      INTEGER RULE, RESTRT,SYMIN,SYMBOL,VARSTR,SYMSTR
      LOGICAL  MATCH
      IF(ITRACE.NE.0) WRITE(6,99)
   99 FORMAT(    30H     SUBROUTINE RERITE CALLED )
      IF(ITRACE.NE.0) WRITE(6,200) ILIST,I
  200 FORMAT(2I6)
      DO 1 L=1,29
    1 SYMIN(L)=ITABLE(ILIST,I,L)
    2 CALL RULENO
      IF(IRULE.EQ.0.AND.SYMIN(23).NE.1) GO TO 3
      CALL RULRIT
      IF(SYMIN(4).EQ.0) RETURN
      GO TO 2
    3 WRITE(6,100) SYMIN(1)
  100 FORMAT(10X,25HNO RULE FOUND FOR SYMBOL-I3)
      STOP
      END
```

Subprogram RULENO

```
      SUBROUTINE RULENO
      COMMON  RULE(800,28),IPSI(100,8,2),RESTRT(5,15),ITABLE(2,50,29),
     1 ILIST,ILIST1,I,J,ITERM,SYMIN(29),IA,IRULE,IKLASS,SYMBOL(29),IB,
     2 IMAX,IEND,VARSTR(29),IVAR(10,29),SYMSTR(10,29),IREG(10,3),MATCH,
     3 ITRACE
      INTEGER RULE,RESTRT,SYMIN,SYMBOL, VARSTR,SYMSTR
      LOGICAL MATCH
C
C *** THIS SUBROUTINE COMPUTES THE  RULE   NUMBER OF THE FIRST AND LAST
C *** SYMBOL OF THE SET OF RULES THAT APPLY TO A GIVEN INPUT SYMBOL
C
      IF(ITRACE.NE.0) WRITE(6,99)
   99 FORMAT(     30H      SUBROUTINE RULENO CALLED )
      IF( SYMIN(4).EQ.0) GO TO 1
      NF=SYMIN(3)
      GO TO(4,5,6,7),NF
    4 IRULE=4
      IMAX =15
      GO TO 3
    5 IRULE=16
      IMAX =27
      GO TO 3
    6 IRULE=28
      IMAX =35
      GO TO 3
    7 IRULE=36
      IMAX =41
      GO TO 3
    1 IF(SYMIN(7).EQ.0.OR.SYMIN(8).EQ.0) GO TO 2
      IRULE=1
      IMAX =3
      GO TO 3
    2 IS=SYMIN(1)
      IC=SYMIN(6)
      IRULE=IPSI(IS,IC,1)
      IMAX =IPSI(IS,IC,2)
    3 CONTINUE
      IF(ITRACE.NE.0) WRITE(6,200) IRULE,IMAX
  200 FORMAT(2I6)
      RETURN
      END
```

Subprogram RULRIT

```
      SUBROUTINE RULRIT

C
C ***   THIS SUBROUTINE REWRITES SYMBOLS UNDER CONTROL OF SUBROUTINE RERITE
C
      COMMON  RULE(800,28),IPSI(100,8,2),RESTRT(5,15),ITABLE(2,50,29),
     1 ILIST,ILIST1,I,J,ITERM,SYMIN(29),IA,IRULE,IKLASS,SYMBOL(29),IB,
     2 IMAX,IEND,VARSTR(29),IVAR(10,29),SYMSTR(10,29),IREG(10,3),MATCH,
     3 ITRACE
      COMMON/INR/  INREAD(29)
      INTEGER RULE,RESTRT,SYMIN,SYMBOL,VARSTR,SYMSTR
      LOGICAL MATCH
      IF(ITRACE.NE.0) WRITE(6,99)
   99 FORMAT(    30H     SUBROUTINE RULRIT CALLED )
      IF(SYMIN(4).EQ.0.AND.SYMIN(8).EQ.0.AND.SYMIN(23).EQ.1) GO TO 13
   12 DO 1 L=1,28
    1 SYMBOL(L)=RULE(IRULE,L)
      IF(ITRACE.NE.0) WRITE(6,200) SYMIN,SYMBOL
  200 FORMAT(29I4)
      CALL SYMACH
      IF(MATCH) CALL VARATT
      IF(MATCH.AND.SYMBOL(27).NE.0) CALL LIMIT
      IF(SYMBOL(27).NE.0) CALL LIMIT
      IF(.NOT.MATCH) GO TO 11
      IEND=IRULE+SYMBOL(26)
      ITABLE(ILIST,I,26)=ITABLE(ILIST,I,26)+SYMBOL(26) -1
    2 IRULE1=IRULE+1
      DO   7 K=IRULE1,IEND
      DO 3  L=1,28
    3 SYMBOL(L)=RULE(K,L)
      IF(SYMIN(23).EQ.1.AND.SYMBOL(1).NE.86)  SYMBOL(23)=1
      IF(SYMBOL(4).EQ.-1) GO TO 8
      CALL INPUT
      IF(.NOT.MATCH) GO TO 10
      IF(SYMBOL(2).EQ.0.AND.RULE(K,2).EQ.9) GO TO  71
      IF(SYMBOL(2).EQ.1.AND.(RULE(K,2).EQ.1.OR.RULE(K,2).EQ.9)) GO TO 5
      WRITE(6,100) SYMBOL,(RULE(K,L),L=1,28)
  100 FORMAT(10X,56HERROR NOTED BETWEEN SYMBOL AND RULE IN SUBROUTINE RU
     1LRIT//10X, 10HSYMBOL IS ,29I4//10X,10HRULE IS   ,28I4//)
      SYMBOL(2)=1
    5 J=J+I
      DO 6 L=1,29
    6 ITABLE(ILIST1,J,L)=SYMBOL(L)
      ITERM=ITERM+SYMBOL(23)
      ITABLE(ILIST1,J,26)=1
      GO TO 7
   71 ITABLE(ILIST,I,26)=ITABLE(ILIST,I,26)-1
    7 CONTINUE
      IF(SYMIN(4).LE.2) SYMIN(4)=0
      RETURN
    8 SYMIN(4)=SYMIN(4)-1
      RETURN
   11 IRULE=IRULE+SYMBOL(26)+1
      ITABLE(ILIST,I,26)=1
      IF(IRULE.LT.IMAX) GO TO 12
      WRITE(6,101) IRULE
  101 FORMAT(10X,87HSUBROUTINE RULRIT FAILED TO FIND APPLICABLE RULE AMO
     2NG SET OF RULES PRECEDING RULE NO. ,I3)
      STOP
   13 J=J+1
      DO 14 L=1,29
   14 ITABLE(ILIST1,J,L)=SYMIN(L)
      ITERM=ITERM+SYMIN(23)
      ITABLE(ILIST1,J,26)=1
      RETURN
   10 WRITE(6,102) SYMBOL(1),INREAD(1)
  102 FORMAT(10X,22HINPUT ERROR AT SYMBOL-,I3,18H, INPUT SYMBOL IS-,I3)
      STOP
      END
```

Subprogram SYMACH

```
      SUBROUTINE SYMACH
C
C *** THIS SUBROUTINE COMPARES A GIVEN SYMBOL WITH THE LEFT  HAND MEMBER
C *** OF A GIVEN RULE. VARIABLE MATCH IS SET TO TRUE IF THE CRITERIA ARE
C *** MET, AND TO FALSE OTHERWISE
C
      COMMON  RULE(800,28),IPSI(100,8,2),RESTRT(5,15),ITABLE(2,50,29),
     1 ILIST,ILIST1,I,J,ITERM,SYMIN(29),IA,IRULE,IKLASS,SYMBOL(29),IB,
     2 IMAX,IEND,VARSTR(29),IVAR(10,29),SYMSTR(10,29),IREG(10,3),MATCH,
     3 ITRACE
      INTEGER RULE,RESTRT,SYMIN,SYMBOL,VARSTR,SYMSTR
      LOGICAL MATCH
      IF(ITRACE.NE.0) WRITE(6,99)
   99 FORMAT(    30H     SUBROUTINE SYMACH CALLED )
      MATCH=.TRUE.
      DO 1 L=1,27
      IF(L.EQ.1) GO TO 6
      IF((L.EQ.3.AND.SYMBOL(1).NE.97).OR.(L.EQ.7.AND.SYMBOL(1).NE.86))
     1  GO TO 1
      IF(SYMBOL(L).GE.0.AND.SYMBOL(L).LE.9) GO TO 3
      GO TO 1
    3 IF(SYMBOL(L).EQ.SYMIN(L)) GO TO 1
      IF((SYMIN(7).EQ.0.OR.SYMIN(8).EQ.0).AND.SYMIN(4).EQ.0) GO TO 5
      GO TO 4
    5 IF(SYMBOL(L).EQ.9.AND.SYMIN(L).EQ.0) GO TO 1
      GO TO 4
    6 IF(SYMBOL(1).EQ.SYMIN(1).OR.SYMBOL(1).EQ.97) GO TO 1
      GO TO 4
    1 CONTINUE
      RETURN
    4 MATCH=.FALSE.
      IF(ITRACE.NE.0) WRITE(6,200) L
  200 FORMAT(5X,  18HMATCH=FALSE FOR L=, I3)
      RETURN
      END
```

Subprogram VARATT

```
      SUBROUTINE VARATT
C
C *** THIS SUBROUTINE COMPUTES VALUES FOR THE DEPENDENT VARIABLES OF A
C *** GIVEN GRAMMAR RULE
C
      COMMON  RULE(800,28),IPSI(100,8,2),RESTRT(5,15),ITABLE(2,50,29),
     1 ILIST,ILIST1,I,J,ITERM,SYMIN(29),IA,IRULE,IKLASS,SYMBOL(29),IB,
     2 IMAX,IEND,VARSTR(29),IVAR(10,29),SYMSTR(10,29),IREG(10,3),MATCH,
     3 ITRACE
      INTEGER RULE, RESTRT,SYMIN,SYMBOL,VARSTR,SYMSTR
      LOGICAL MATCH
      IF(ITRACE.NE.0) WRITE(6,99)
   99 FORMAT(    30H     SUBROUTINE VARATT CALLED )
      DO 1 L=1,29
    1 VARSTR(L)=0
      IF(SYMIN(4).EQ.0.AND.SYMIN(8).EQ.0) GO TO 2
      SYMBOL(1)=SYMIN(1)
      VARSTR(1)=SYMIN(1)
    2 DO 3 L=2,22
      IF(SYMBOL(L).GE.0.AND.SYMBOL(L).LE.9) GO TO 3
      K=SYMIN(L)
      IF(SYMBOL(L).LT.0)  K=IABS(SYMBOL(L))
      VARSTR(L)=K
      SYMBOL(L)=K
    3 CONTINUE
      DO 4 L=23,29
    4 VARSTR(L)=SYMIN(L)
      IF(ITRACE.NE.0) WRITE(6,200) VARSTR, SYMBOL
  200 FORMAT(29I4)
      RETURN
      END
```

Subprogram INFLEX

```
      SUBROUTINE INFLEX(SYMBOL,WORD,NC)
C
C *** THIS SUBROUTINE COMPUTES THE CORRECT ORTHOGRAPHY FOR MODERN HEBREW WORDS
C
      COMMON/ABC/ TRANSL(50)
      DIMENSION SYMBOL(29),WORD(19),W(4),ATRAN(22),AWORD(19)
      INTEGER SYMBOL,A,B,N,G,P,M,H,T,W, ATRAN,WORD,F,X,R
      DATA ATRAN/4,4,7,7,2,2,7,4,4,7,4,6,6,7,5,7,7,1,3,3,8,6/
      ICLASS=SYMBOL(6)
      A=SYMBOL(1)
      B=SYMBOL(18)
      N=SYMBOL(10)
      G=SYMBOL(11)
      P=SYMBOL(12)
      R=SYMBOL(13)
      M=SYMBOL(16)
      T=SYMBOL(17)
      J=SYMBOL(23)
      W(1)=SYMBOL(19)
      W(2)=SYMBOL(20)
      W(3)=SYMBOL(21)
      W(4)=SYMBOL(22)
      H=SYMBOL(24)
      X=SYMBOL(29)
      IF((A.EQ.82.OR.A.EQ.84.OR.A.EQ.85.OR.A.EQ.96.OR.A.EQ.98).AND.
     1 J.EQ.1)  J=2
      DO 1 L=1,19
    1 WORD(L)=44
      IF(M.EQ.3) M=1
C
C ***                 COMPUTE THE VALUE OF F
C
      F=3
      IF((A.EQ.79.AND.ICLASS.EQ.1).OR.A.EQ.83.OR.(A.EQ.89.AND.R.
     2 LT.5).OR.(A.EQ.90.AND.ICLASS.EQ.1).OR.(A.EQ.92.AND.ICLASS.EQ.1))
     3 F=2
      IF((A.EQ.91.AND.ICLASS.EQ.3).OR.(A.EQ.92.AND.ICLASS.GT.1).OR.
     2 (A.EQ.93.AND.ICLASS.EQ.7))  F=1
C
C ***           READ IN NON-HEBREW WORDS
C
      IF(A.NE.87.OR.ICLASS.NE.3) GO TO 111
      READ(5,112) (AWORD(K),K=2,18)
  112 FORMAT(17A1)
      DO 113 K=2,18
      CALL ALPHAI(AWORD(K),K1)
  113 WORD(K)=K1
      GO TO 361
  111 CONTINUE
```

```
C
C ***                 COMPUTE VALUE OF WORD TYPE(A) AND STATE(J)
C
      A=A-76
      A=ATRAN(A)
      GO TO (3,3,3,3,101,103,102),A
  101 GO TO(210,220,230,240,102),ICLASS
  102 J=1
  103 N=1
      G=1
      A=4
      GO TO 3
C
C ***             COMPUTE ORTHOGRAPHY OF DEMONSTRATIVE PRONOUN
C
  210 IF(N.GT.1) GO TO 212
      WORD(2)=16
      IF(G.EQ.2) GO TO 211
      WORD(3)=14
      GO TO 361
  211 WORD(3)=15
      WORD(4)=31
      GO TO 361
  212 WORD(2)=10
      WORD(3)=21
      WORD(4)=14
      GO TO 361
C
C ***                 COMPUTE ORTHOGRAPHY OF PERSONAL PRONOUNS(NOMINATIVE)
C
  220 WORD(2)=10
      IF(P.EQ.3) WORD(2)=14
      IF(P.EQ.1) WORD(3)=23
      IF(N.EQ.1.AND.G.EQ.1.AND.P.EQ.3)  WORD(3)=15
      IF(N.EQ.1.AND.G.EQ.2.AND.P.EQ.3)  WORD(3)=19
      IF(P.EQ.2) WORD(4)=31
      IF(N.EQ.1.AND.P.EQ.1)WORD(4)=19
      IF(N.EQ.1.AND.P.EQ.3)WORD(4)=10
      IF(N.GT.1.AND.P.EQ.1)WORD(4)=17
      IF(N.EQ.1.AND.G.EQ.1.AND.P.EQ.2)  WORD(5)=14
      IF(N.GT.1.AND.(P.EQ.1.OR.(G.EQ.2.AND.P.GT.1)))   WORD(5)=23
      IF(N.GT.1.AND.G.EQ.1.AND.P.GT.1)  WORD(5)=22
      IF(N.GT.1.AND.P.EQ.1)    WORD(6)=15
      GO TO 361
C
C ***                 COMPUTE ORTHOGRAPHY OF PERSONAL PRONOUNS(ACCUSATIVE)
C
  230 IF(H.EQ.1) WORD(2)=19
      IF((N.EQ.1.AND.((P.EQ.1.AND.(J.EQ.2.OR.J.EQ.5.OR.J.EQ.6)).OR.(P.
     2 EQ.3.AND.J.EQ.6))).OR.(N.GT.1.AND.P.EQ.1))  WORD(3)=23
```

```
      IF(P.EQ.3.AND.((N.EQ.1.AND.G.EQ.1.AND.(J.EQ.2.OR.J.EQ.5)).OR.(N.
     2 GT.1.AND.(J.EQ.3.OR.J.EQ.5.OR.J.EQ.6))))   WORD(3)=14
      IF(P.EQ.2) WORD(3)=20
      IF(N.EQ.1.AND.P.EQ.1)   WORD(4)=19
      IF(N.GT.1.AND.P.EQ.1)   WORD(4)=15
      IF(N.EQ.1.AND.G.EQ.1.AND.P.EQ.3)  WORD(4)=15
      IF(N.EQ.1.AND.G.EQ.2.AND.P.EQ.3)  WORD(4)=14
      IF(N.GT.1.AND.G.EQ.1.AND.P.GT.1)  WORD(4)=22
      IF(N.GT.1.AND.G.EQ.2.AND.P.GT.1)  WORD(4)=23
      GO TO 361
C
C ***                COMPUTE ORTHOGRAPHY OF RELATIVE PRONOUN
C
  240 WORD(2)=30
      F=2
      GO TO 361
C
C ***                       COMPUTE NUMBER-GENDER TRANSFORATION
C
    3 IF(X.EQ.0) GO TO 59
      GO TO(41,42,43,44,45,46,47,48,49,50,51),X
   41 IF(N.EQ.1.AND.G.EQ.1)  G=2
      GO TO 59
   42 IF(N.EQ.1.AND.G.EQ.2)  G=1
      GO TO 59
   43 IF(N.GT.1.AND.G.EQ.1)  G=2
      GO TO 59
   44 IF(N.GT.1.AND.G.EQ.2)  G=1
      GO TO 59
   45 IF(G.EQ.1) G=2
      GO TO 59
   46 IF(G.EQ.2) G=1
      GO TO 59
   47 G=1
      N=1
      GO TO 59
   48 G=2
      N=1
      GO TO 59
   49 G=1
      N=3
      GO TO 59
   50 G=2
      N=3
   51 N=2
   59 CONTINUE
C
C ***                       COMPUTE VALUE OF B(A,N,G,P,T,M)
C
      IF(A.EQ.1.AND.T.EQ.2.AND.M.NE.2) GO TO 4
      WORD(2)=44
      GO TO 5
      4 IF(G.EQ.1.AND.P.EQ.3) WORD(2)=19
        IF(G.EQ.2.AND.P.EQ.3) WORD(2)=31
        IF(P.EQ.2)            WORD(2)=31
        IF(N.EQ.1.AND.P.EQ.1) WORD(2)=10
        IF(N.NE.1.AND.P.EQ.1) WORD(2)=23
C
C ***                  COMPUTE VALUE OF S1(A,B,T,M)
C
      5 WORD(3)=44
        GO TO(6,7,8,9),A
      6 IF(B.EQ.3.AND.T.EQ.1.AND.M.EQ.1)               WORD(3)=23
        IF(B.EQ.3.AND.T.EQ.2.AND.M.EQ.2)               WORD(3)=14
        IF(((B.GT.5.AND.B.LT.9).OR.B.EQ.10).AND.T.EQ.1) WORD(3)=14
        GO TO 10
      7 IF(B.GT.3.AND.B.LT.11)                         WORD(3)=22
        IF(B.EQ.3)                                     WORD(3)=23
        GO TO 10
      8 IF((B.GT.5.AND.B.LT.9).OR.B.EQ.10)                  WORD(3)=14
        IF(B.EQ.3.AND.J.EQ.1) WORD(3)=23
        IF(B.EQ.3.AND.J.GT.1) WORD(3)=14
        GO TO 10
      9 IF(B.EQ.7.OR.B.EQ.8)                           WORD(3)=10
        IF(B.GT.8.AND.B.LT.12)                         WORD(3)=14
        IF(B.GT.11.AND.B.LT.15)                        WORD(3)=22
        IF(B.GT.14.AND.B.LT.18)                        WORD(3)=31
        IF(B.EQ.18.OR.B.EQ.20)                         WORD(3)=23
        IF(B.EQ.21)                                    WORD(3)=30
C
C ***                  COMPUTE VALUE OF S2(A,B,N,  P,T,M,W,J)
C
     10 WORD(4)=44
        IF(A.GT.3) GO TO 12
        IF((A.GT.0.AND.A.LT.4).AND.(B.EQ.6.OR.B.EQ.10).AND.W(1).NE.24.
       2 AND.W(1).NE.30.AND.W(1).NE.27.AND.W(1).NE.16.AND.W(1).NE.31.AND.
       3  W(1).NE.13.AND.W(1).NE.18) WORD(4)=31
        IF((A.GT.0.AND.A.LT.4).AND.B.EQ.8)      WORD(4)=15
        IF((B.EQ.6.OR.B.EQ.10).AND.(W(1).EQ.13.OR.W(1).EQ.18.OR.W(1).
       1 EQ.31).AND.(N.NE.1.OR.P.NE.1.OR.T.NE.2)) WORD(4)=19
        IF(A.NE.1) GO TO 11
        IF(B.EQ.12.AND.T.EQ.2.AND.M.EQ.1.AND.W(1).EQ.10.AND.
       1 N.EQ.1.AND.P.EQ.1)  WORD(4)=15
        IF(B.EQ.3.AND.T.EQ.2.AND.W(1).NE.10.AND.(N.NE.1.OR.P.NE.1))
       2 WORD(4)=19
        IF(B.EQ.12.AND.T.EQ.2.AND.M.EQ.1.AND. W(2).EQ.15) WORD(4)=19
        IF(B.EQ.3.AND.T.EQ.2.AND.(W(1).EQ.15.OR.(W(2).EQ.W(3).AND.W(4).EQ
       2 44)))  WORD(4)=19
        IF((B.EQ.11.OR.B.EQ.13).AND.N.EQ.1.AND.P.EQ.1.AND.T.EQ.2.AND.
       2 M.EQ.1.AND.(W(1).EQ.10))  WORD(4)=19
      IF((B.EQ.11.OR.B.EQ.13).AND.(N.NE.1.OR.P.NE.1).AND.T.EQ.2.AND.M.
     2 EQ.1.AND.W(2).EQ.W(3).AND.W(4).EQ.44)   WORD(4)=19
      IF(B.GT.10.AND.B.LT.15.AND.T.EQ.2.AND.M.EQ.1.AND.W(1).EQ.23.
     2 AND.(N.NE.1.OR.P.NE.1)) WORD(4)=19
      GO TO 12
   11 IF(A.EQ.3.AND.B.EQ.3.AND.W(1).NE.10) WORD(4)=19
      IF(A.EQ.2.AND.B.EQ.3.AND.W(1).EQ.23) WORD(4)=19
C
C ***                  COMPUTE VALUE OF S3(A,B,N,G,P,M,T,W,J)
C
   12 WORD(5)=W(1)
      IF(A.GT.4) GO TO 14
      IF(A.EQ.4.AND.B.EQ.25.AND.G.EQ.1)  WORD(5)=44
      IF(A.LE.3              .AND.B.EQ.8.AND.(W(1).EQ.19.OR.W(1).EQ.15.
     2OR.W(1).EQ.23))    WORD(5)=44
      IF(A.EQ.1.AND.B.GT.10.AND.B.LT.14.AND.N.EQ.1.AND.P.EQ.1.AND.
     2 M.EQ.1.AND.T.EQ.2.AND.W(1).EQ.10) WORD(5)=44
      IF(W(1).NE.23) GO TO 14
      IF(A.NE.1) GO TO 13
      IF((B.EQ.3.OR.B.EQ.7).AND.M.EQ.1.AND.T.EQ.1) WORD(5)=19
      IF(B.EQ.7.AND.T.EQ.2)                        WORD(5)=44
      IF(B.EQ.7.AND.M.EQ.2.AND.T.EQ.2.AND.W(3).EQ.14.AND.W(4).EQ.44)
     2 WORD(5)=19
      IF(B.GT.10.AND.B.LT.14.AND.T.EQ.2)           WORD(5)=44
      IF(B.EQ.14.AND.M.EQ.1.AND.T.EQ.2)            WORD(5)=44
   13 IF(A.EQ.2.AND.B.EQ.3)                        WORD(5)=44
      IF((A.EQ.2.OR.A.EQ.3).AND.B.EQ.7)            WORD(5)=44
      IF(A.EQ.3.AND.B.EQ.13.AND.J.GT.1)            WORD(5)=44
C
C ***                  COMPUTE VALUE OF S4(A,B,W)
C
   14 WORD(6)=44
      IF(A.GT.3.OR.(B.NE.6.AND.B.NE.10)) GO TO 15
      IF(W(1).EQ.24.OR.W(1).EQ.30) WORD(6)=31
      IF(W(1).EQ.27)               WORD(6)=18
      IF(W(1).EQ.16)               WORD(6)=13
C
C ***                  COMPUTE VALUE OF S5(A,B,M,T,W,J)
C
   15 WORD(7)=44
      IF(A.GT.3) GO TO 17
      IF((B.EQ.4.OR.B.EQ.6).AND.W(2).EQ.19)  WORD(7)=19
      IF((B.EQ.4.OR.B.EQ.6).AND.T.EQ.2.AND.W(2).EQ.15) WORD(7)=15
      IF(B.EQ.5)  WORD(7)=15
      IF(B.EQ.9.OR.B.EQ.10)  WORD(7)=15
      IF(A.GT.1) GO TO 16
      IF(B.EQ.4.AND.M.EQ.1.AND.T.EQ.1)        WORD(7)=19
   16 IF(A.EQ.2.AND.B.EQ.13.AND.W(1).EQ.23)             WORD(7)=15
      IF(A.EQ.2.AND.B.EQ.12.AND.W(2).NE.15.AND.W(2).NE.19) WORD(7)=15
      IF(A.EQ.2.AND.B.EQ.14.AND.W(2).NE.15.AND.W(2).NE.19) WORD(7)=15
      IF( A.EQ.2             .AND.B.EQ.12.AND.W(2).EQ.W(3).AND.
     2 W(4).EQ.44) WORD(7)=15
```

```
      IF(A.EQ.3.AND.B.EQ.3.AND.W(2).EQ.W(3).AND.W(4).EQ.44.AND.
     1 J.EQ.1)  WORD(7)=15
   17 IF(A.EQ.4.AND.B.EQ.4)   WORD(7)=15
      IF(A.EQ.4.AND.(B.EQ.5.OR.B.EQ.6)) WORD(7)=19
C
C ***                  COMPUTE VALUE OF S6(A,B,N,G,P,T,M,W,J)
C
   18 WORD(8)=W(2)
      IF(A.EQ.1.AND.B.EQ.12.AND.T.EQ.2.AND.W(2).EQ.15) WORD(8)=44
      IF(A.EQ.1.AND.B.EQ.14.AND.N.GT.1.AND.G.EQ.2.AND.P.GT.1.AND.T.EQ.
     2 2.AND.W(2).EQ.19) WORD(8)=44
      IF(A.LT.4.AND.(B.EQ.9.OR.B.EQ.10))      WORD(8)=W(3)
      IF( A.LT.4.AND.(B.EQ.7.OR.B.EQ.8).AND.(W(2).EQ.15.OR.
     1 W(2).EQ.19.OR.(W(2).EQ.W(3).AND.W(4).EQ.44))) WORD(8)=44
      IF(B.EQ.3.AND.W(2).EQ.W(3).AND.W(4).EQ.44) WORD(8)=44
      IF(A.EQ.1.AND.B.GT.12.AND.B.LT.15.AND.T.EQ.1.AND.M.EQ.1.AND.
     2 (W(2).EQ.15.OR.W(2).EQ.19))  WORD(8)=44
      IF(A.EQ.1.AND.B.GT.10.AND.B.LT.15.AND.W(2).EQ.W(3).AND.W(4).EQ.44)
     2 WORD(8)=44
      IF(A.EQ.1.AND.B.EQ.14.AND.T.EQ.2.AND.M.EQ.2.AND.W(2).EQ.19)
     2 WORD(8)=44
      IF( A.EQ.2              .AND.B.GT.11.AND.B.LT.15.AND.(W(2).EQ.15.
     2 OR.W(2).EQ.19))   WORD(8)=44
      IF(A.EQ.3.AND.B.GT.11.AND.B.LT.15.AND.W(2).EQ.19) WORD(8)=44
      IF(A.EQ.3.AND.B.GT.11.AND.B.LT.15.AND.W(2).EQ.15.AND.J.GT.1)
     1   WORD(8)=44
      IF( A.EQ.2              .AND.(B.EQ.11.OR.B.EQ.13).AND.W(2).EQ.W(3).
     2 AND.W(4).EQ.44) WORD(8)=44
      IF(A.NE.4) GO TO 19
      IF((B.EQ.24.OR.B.EQ.29.OR.B.EQ.35).AND.((N.GT.1.AND.G.EQ.1).OR.
     1 (N.EQ.3.AND.G.EQ.2)))  WORD(8)=44
      IF(B.EQ.25.AND.((G.EQ.1.AND.J.EQ.1).OR.J.GT.1))  WORD(8)=44
      IF(B.EQ.28.AND.N.EQ.1.AND.G.EQ.1.AND.J.EQ.1)  WORD(8)=44
      IF(B.EQ.30.AND.N.LT.3.AND.G.EQ.2)  WORD(8)=44
      IF(B.EQ.38.AND.N.EQ.1.AND.J.EQ.1)  WORD(8)=44
C
C ***                  COMPUTE VALUE OF S7(A,B,N,G,P,T,M,J,W)
C
   19 WORD(9)=44
      IF(A.NE.1) GO TO 20
      IF(W(2).EQ.W(3).AND.W(4).EQ.44.AND.T.EQ.2.AND.((M.EQ.2.AND.B.GT.
     2 10.AND.B.LT.15).OR.(M.EQ.1.AND.(B.EQ.11.OR.B.EQ.14))))WORD(9)=15
      IF(B.EQ.11.AND.((N.EQ.1.AND.G.EQ.1.AND.P.NE.1).OR.(N.EQ.1.AND.
     2 G.EQ.2.AND.P.EQ.2).OR.(N.GT.1.AND.P.EQ.1)).AND.T.EQ.1.AND.M.EQ.
     3 1.AND. W(3).NE.14.AND.W(2).NE.15.AND.W(2).NE.19.AND.W(2).NE.
     4 W(3))   WORD(9)=15
      IF(B.EQ.14.AND.((N.EQ.1.AND.P.EQ.3.AND.T.EQ.2.AND.M.EQ.1).OR.
     2 (P.EQ.1.AND.T.EQ.2.AND.M.EQ.1).OR.(N.GT.1.AND.G.EQ.2.AND.P.GT.1.
     3 AND.T.EQ.2).OR.(N.EQ.1.AND.G.EQ.1.AND.P.EQ.2.AND.T.EQ.2)).AND.
     4 J.EQ.1.AND.W(3).NE.14.AND.W(2).NE.15.AND.W(2).NE.19.AND.W(2).
     5 NE.W(3).AND.W(1).NE.23)      WORD(9)=15
```

```
      IF(B.EQ.14.AND.((N.EQ.1.AND.G.EQ.1).CR.(N.EQ.2.AND.G.EQ.2)).AND.
     2 P.EQ.2.AND.T.EQ.2.AND.M.EQ.2.AND.J.GT.1.AND.W(1).EQ.23)
     3 WORD(9)=15
      IF(B.EQ.14.AND.((N.EQ.1.AND.P.EQ.3).OR.P.EQ.1.OR.(N.GT.1.AND.G.EQ.
     2 2.AND.P.GT.1).OR.(N.EQ.1.AND.G.EQ.1.AND.P.EQ.2)).AND.T.EQ.2.AND.
     3 M.EQ.1.AND.J.EQ.1.AND.W(1).EQ.23) WORD(9)=15
      IF(B.EQ.7.AND.(W(2).EQ.15.OR.W(2).EQ.19)) WORD(9)=19
      IF(B.EQ.7.AND.((P.EQ.3.AND.T.EQ.1.AND.M.EQ.1).OR.T.EQ.2).AND.W(3).
     2 NE.14.AND.W(3).NE.17) WORD(9)=19
      IF(B.EQ.7.AND.W(2).EQ.W(3).AND.W(4).EQ.44.AND.((T.EQ.1.AND.P.LT.
     2 3).OR.(N.NE.1.OR.P.NE.1))) WORD(9)=19
      IF(B.EQ.7.AND.T.EQ.2.AND.W(3).EQ.17.AND.(N.EQ.1.OR.G.NE.2.OR.P.
     2 EQ.1)) WORD(9)=19
   20 IF(A.NE.2.AND.A.NE.3) GO TO 21
      IF(W(2).NE.15.AND.W(2).NE.19.AND.W(2).NE.W(3).AND.W(3).NE.14.AND.
     1 B.EQ.11.AND.((A.EQ.2.AND.N.EQ.1.AND.G.EQ.1).OR.A.EQ.3))WORD(9)=15
      IF(B.EQ.2.AND.W(2).NE.15)    WORD(9)=15
      IF(B.EQ.7.AND.W(3).NE.14.AND.                           (N.NE.1.OR.
     2 G.NE.2.OR.H.NE.1).AND.A.EQ.2.AND.W(2).NE.W(3)) WORD(9)=19
      IF(A.EQ.2.AND.B.EQ.7.AND.W(2).EQ.W(3).AND.W(4).EQ.44.AND.(N.NE.1.
     2  OR.G.NE.1).AND.(N.NE.1.OR.G.NE.2.OR.H.NE.1)) WORD(9)=19
   21 IF(A.NE.3) GO TO 22
      IF(B.EQ.7.AND.W(3).NE.14.AND.W(2).NE.W(3)) WORD(9)=19
      IF(B.EQ.6.AND.J.EQ.1.AND.W(3).EQ.14.AND.W(4).EQ.44)   WORD(9)=15
      IF(B.GE.12.AND.B.LE.14 .AND.W(3).NE.14.AND.W(2).NE.15.AND.W(2).
     2 NE.19)  WORD(9)=15
      IF((B.EQ.3.OR.B.EQ.5.OR.B.EQ.9).AND.J.EQ.1)  WORD(9)=15
      IF(B.GE.11.AND.B.LE.14.AND.J.EQ.1.AND.W(2).EQ.W(3).AND.W(4).EQ.
     1 44) WORD(9)=15
      IF(J.GT.1.AND.W(3).EQ.14.AND.W(4).EQ.44)  WORD(9)=15
   22 IF(A.EQ.4.AND.(B.EQ.2.OR.B.EQ.10.OR.B.EQ.13.OR.B.EQ.16))WORD(9)=19
      IF(A.EQ.4.AND.(B.EQ.3.OR.B.EQ.6.OR.B.EQ.8.OR.B.EQ.11.OR.B.EQ.14.
     2 OR.B.EQ.17.OR.B.EQ.20)) WORD(9)=15
C
C ***                   COMPUTE VALUE OF S8(A,B,N,G,P,T,M,J,W)
C
   23 WORD(10)=W(3)
      IF(A.NE.1) GO TO 24
      IF(B.NE.9.AND.B.NE.10.AND.N.EQ.1.AND.G.EQ.2.AND.P.EQ.3.AND.T.EQ.1.
     2 AND.M.EQ.1.AND.W(3).EQ.14.AND.W(4).EQ.44) WORD(10)=31
      IF(B.GT.2.AND.B.LT.9.AND.                 (((P.EQ.1.OR.P.EQ.2).AND.T.EQ.
     2 1.AND.M.EQ.1).OR.(N.GT.1.AND.G.EQ.2.AND.(P.EQ.2.OR.P.EQ.3).AND.
     3 T.EQ.2)).AND.W(3).EQ.14.AND.W(4).EQ.44)    WORD(10)=19
      IF(B.GT.10.AND.B.LT.15.AND.(((P.EQ.1.OR.P.EQ.2).AND.T.EQ.1).OR.
     2 (N.GT.1.AND.G.EQ.2.AND.P.GT.1.AND.T.EQ.2)).AND.M.EQ.1.AND.W(3).
     3 EQ.14.AND.W(4).EQ.44)  WORD(10)=19
      IF(B.GT.10.AND.B.LT.15.AND.N.GT.1.AND.G.EQ.2.AND.P.EQ.2.AND.T.EQ.
     2 2.AND.M.EQ.2.AND.J.GT.1.AND.W(3).EQ.14.AND.W(4).EQ.44)
     3 WORD(10)=19
      IF(B.GT.10.AND.B.LT.15.AND.((N.EQ.1.AND.P.EQ.3).OR.(N.EQ.1.AND.
     2 G.EQ.1.AND.P.EQ.2).OR.P.EQ.1).AND.T.EQ.2.AND.M.EQ.1.AND.J.GT.1.
     3 AND.W(1).EQ.23.AND.W(3).EQ.14.AND.W(4).EQ.44) WORD(10)=19
      IF((B.EQ.3.OR.B.EQ.7.OR.B.EQ.8.OR.(B.GT.10.AND.B.LT.15)).AND.
     2 (P.EQ.1.OR.P.EQ.2).AND.T.EQ.1.AND.M.EQ.1.AND.W(1).EQ.23.AND.
     3 W(3).EQ.23.AND.W(4).EQ.44)  WORD(10)=44
      IF(((B.GT.2.AND.B.LT.9).OR.(B.GT.10.AND.B.LT.15)).AND.((N.GT.1.
     2 AND.P.EQ.3.AND.T.EQ.1.AND.M.EQ.1).OR.(N.EQ.1.AND.G.EQ.2.AND.P.
     3 EQ.2.AND.T.EQ.2).OR.(N.GT.1.AND.G.EQ.1.AND.(P.EQ.2.OR.P.EQ.3).
     4 AND.T.EQ.2)).AND.W(3).EQ.14.AND.W(4).EQ.44)      WORD(10)=44
      IF(((B.GT.2.AND.B.LT.9).OR.(B.GT.10.AND.B.LT.15)).AND.((N.EQ.1.
     2 AND.G.EQ.1.AND.P.EQ.3.AND.T.EQ.1.AND.M.EQ.1).OR.(N.EQ.1.AND.P.
     3 EQ.3.AND.T.EQ.2.AND.M.EQ.1).OR.(N.EQ.1.AND.G.EQ.1.AND.P.EQ.2.AND.
     4 T.EQ.2).OR.(P.EQ.1.AND.T.EQ.2.AND.M.EQ.1)).AND.J.GT.1.AND.W(3).
     5 EQ.14.AND.W(4).EQ.44)  WORD(10)=44
      GO TO 28
   24 IF( A.NE.2) GO TO 25
      IF(B.GT.2                           .AND.J.GT.2.AND.W(3).EQ.14.AND.
     2 W(4).EQ.44) WORD(10)=44
      IF(B.GT.2.AND.B.LT.15.AND.
     1((N.EQ.1.AND.G.EQ.2).OR.N.GT.1).AND.W(3).EQ.14.AND.W(4).EQ.44)
     2 WORD(10)=44
   25 IF(A.NE.2.AND.A.NE.3) GO TO 26
      IF(B.EQ.2.AND.W(3).EQ.14.AND.W(4).EQ.44)    WORD(10)=15
   26 IF(A.NE.3) GO TO 27
      IF(  J.GT.1.AND.W(3).EQ.14.AND.W(4).EQ.44)  WORD(10)=31
      IF(B.EQ.13.AND.W(1).EQ.23.AND.W(3).EQ.23.AND.W(4).EQ.44)
     2 WORD(10)=44
      GO TO 28
   27 IF(A.NE.4) GO TO 28
      IF((B.EQ.24.OR.B.EQ.34).AND.((N.EQ.1.AND.G.EQ.1).OR.(N.LT.3.AND.
     1 G.EQ.2)))  WORD(10)=44
      IF((B.EQ.26.OR.B.EQ.36).AND.((N.GT.1.AND.G.EQ.1).OR.(N.EQ.3.AND.
     1 G.EQ.2)))  WORD(10)=44
      IF(B.EQ.25.AND.G.EQ.2)  WORD(10)=44
      IF(B.EQ.27.AND..NOT.(N.EQ.1.AND.J.GT.1))  WORD(10)=44
      IF(B.EQ.28.AND..NOT.(N.EQ.3.AND.G.EQ.2)) WORD(10)=44
      IF(B.EQ.29.AND.(J.GT.1.OR.(J.EQ.1.AND.((N.GT.1.AND.G.EQ.1).OR.
     1 (N.EQ.3.AND.G.EQ.2))))) WORD(10)=44
      IF(B.EQ.30) WORD(10)=44
      IF((N.NE.1.OR.G.NE.1).AND.B.NE.34.AND.W(3).EQ.14.AND.W(4).
     1  EQ.44)   WORD(10)=44
      IF(B.NE.34.AND.N.EQ.1.AND.G.EQ.1.AND.J.GT.2.AND.W(3).EQ.14.AND.
     2 W(4).EQ.44)  WORD(10)=44
      IF(B.EQ.38.AND.N.EQ.1.AND.J.EQ.1) WORD(10)=44
      IF(B.EQ.39.AND.G.EQ.2) WORD(10)=44
C
C ***                  COMPUTE VALUE OF S9(A,B,N,G,P,T,M,J,W)
C
   28 WORD(11)=44
C
C ***                  COMPUTE VALUE OF S10(A,B,N,G,P,T,M,J,W)
C
   29 WORD(12)=W(4)
      IF(A.NE.4) GO TO 30
      IF((B.EQ.26.OR.B.EQ.34).AND.((N.EQ.1.AND.G.EQ.1).OR.(N.LT.3.AND.
     1 G.EQ.2)))  WORD(12)=44
      IF(B.EQ.28.AND..NOT.(N.EQ.1.AND.G.EQ.1.AND.J.EQ.1)) WORD(12)=44
      IF(B.EQ.30) WORD(12)=44
      IF(B.EQ.33.AND.N.GT.1.AND.G.EQ.1) WORD(12)=44
      IF((N.NE.1.OR.G.NE.1).AND.B.NE.34.AND.W(4).EQ.14)  WORD(12)=44
      IF(B.NE.34.AND.N.EQ.1.AND.G.EQ.1.AND.J.GT.2.AND.W(4).EQ.14)
     2 WORD(12)=44
      IF(B.EQ.38.AND.J.GT.1) WORD(12)=44
C
C ***                     COMPUTE VALUE OF S11(A,B,N,G,P,T,M,C,W)
C
   30 WORD(13)=44
      IF(A.NE.1) GO TO 31
      IF( B.EQ.3            .AND.(P.EQ.2.OR.P.EQ.1).AND.T.EQ.1.
     2 AND.M.EQ.1.AND.(W(2).EQ.15.OR.W(2).EQ.19)) WORD(13)=15
      IF((B.EQ.3.OR.B.EQ.7.OR.B.EQ.8.OR.(B.GT.10.AND.B.LT.15)).AND.(P.EQ
     2 .1.OR.P.EQ.2).AND.T.EQ.1.AND.M.EQ.1.AND.W(2).EQ.W(3).AND.W(4).
     3 EQ.44)   WORD(13)=15
      IF((B.EQ.3.OR.B.EQ.7.OR.B.EQ.8.OR.B.EQ.14).AND.N.GT.1.AND.G.EQ.2.
     2 AND.(P.EQ.2.OR.P.EQ.3).AND.T.EQ.2.AND.W(2).EQ.W(3).AND.W(4).
     3 EQ.44)   WORD(13)=19
      IF((B.EQ.11.OR.B.EQ.12).AND.N.GT.1.AND.G.EQ.2.AND.P.EQ.2.AND.T.
     2 EQ.2.AND.M.EQ.2.AND.W(2).EQ.W(3).AND.W(4).EQ.44) WORD(13)=19
   31 IF(A.EQ.2.AND.B.EQ.2.AND.W(3).EQ.14.AND.W(4).EQ.44) WORD(13)=19
      IF(A.EQ.3.AND.B.EQ.13.AND.J.GT.1.AND.W(1).EQ.23.AND.
     2 (.NOT.(W(3).EQ.14.AND.W(4).EQ.44))) WORD(13)=31
      IF(A.NE.4) GO TO 32
      IF(B.EQ.23) WORD(13)=15
      IF(B.EQ.19) WORD(13)=19
      IF(B.EQ.31.AND.G.EQ.2)  WORD(13)=19
      IF(B.EQ.32.AND.G.EQ.2)  WORD(13)=15
C
C ***                     COMPUTE VALUE OF S12(A,B,N,G)
C
   32 WORD(14)=44
      IF(A.NE.4) GO TO 33
      IF(B.EQ.22.OR.B.EQ.23)  WORD(14)=23
      IF(B.EQ.32.AND.N.EQ.3.AND.G.EQ.2) WORD(14)=19
C
C ***                     COMPUTE VALUE OF A1(A,N,G)
C
   33 WORD(15)=44
      IF(A.GE.2.AND.A.LE.4.AND.N.EQ.2.AND.G.EQ.2) WORD(15)=31
C
C ***                     COMPUTE VALUE OF A2(A,N,J)
C
      WORD(16)=44
      IF(A.GE.2.AND.A.LE.4.AND.N.EQ.2.AND.J.EQ.1)  WORD(16)=19
C
C ***                     COMPUTE VALUE OF A3(A,N,G,P,T,J)
C
   34 WORD(17)=44
      IF(A.GT.4) GO TO 35
      IF(A.EQ.1.AND.T.EQ.1.AND.(P.EQ.2.OR.(N.EQ.1.AND.P.EQ.1)))
     2 WORD(17)=31
      IF(A.EQ.1.AND.N.GT.1.AND.((P.EQ.1.AND.T.EQ.1).OR.(G.EQ.2.AND.P.GT.
     2 1.AND.T.EQ.2.AND.J.EQ.1)))  WORD(17)=23
      IF(A.EQ.1.OR.A.GT.4) GO TO 35
      IF(N.EQ.2.OR.(N.EQ.3.AND.G.EQ.1)) WORD(17)=19
      IF(N.EQ.3.AND.G.EQ.2)  WORD(17)=15
C
C ***                    COMPUTE VALUE OF A4(A,N,G,P,T,J,H)
C
   35 WORD(18)=44
      IF(A.GT.4) GO TO 36
      IF(A.GT.1) GO TO 352
      IF(N.GT.1) GO TO 351
      IF(G.EQ.2.AND.P.EQ.3.AND.T.EQ.1.AND. J.GT.1) WORD(18)=31
      IF(G.EQ.2.AND.P.EQ.3.AND.T.EQ.1.AND.J.EQ.1)            WORD(18)=14
      IF(G.EQ.2.AND.P.EQ.2.AND.(T.EQ.2.OR.(T.EQ.1.AND.J.GT.1)))
     1 WORD(18)=19
      IF(P.EQ.1.AND.T.EQ.1) WORD(18)=19
      GO TO 36
  351 IF(G.EQ.2.AND.P.GT.1.AND.T.EQ.2.AND.J.EQ.1) WORD(18)=14
      IF(G.EQ.1.AND.P.EQ.2.AND.T.EQ.1.AND.J.EQ.1) WORD(18)=22
      IF(G.EQ.2.AND.P.EQ.2.AND.T.EQ.1.AND.J.EQ.1) WORD(18)=23
      IF(P.GT.1.AND.T.EQ.2.AND.(G.EQ.1.OR.(G.EQ.2.AND.J.GT.1)))
     1 WORD(18)=15
      IF(P.EQ.2.AND.T.EQ.1.AND.J.GT.1) WORD(18)=15
      IF(P.NE.2.AND.T.EQ.1)  WORD(18)=15
      GO TO 36
  352 IF(N.EQ.1.AND.G.EQ.2.AND.J.EQ.1.AND.H.EQ.0) WORD(18)=14
      IF(G.EQ.2.AND.(N.EQ.3.OR.(N.EQ.1.AND.(J.GT.1.OR.(J.EQ.1.AND.
     1 H.EQ.1))))) WORD(18)=31
      IF(J.EQ.1.AND.(N.EQ.2.OR.(N.EQ.3.AND.G.EQ.1))) WORD(18)=22
C
C ***     CHECK FOR 3N/3T ASSIMILATION
C
   36 IF(WORD(13).NE.44.OR.A.NE.1) GO TO 361
      IF(WORD(10).EQ.23.AND.WORD(12).EQ.44.AND.WORD(17).EQ.23)
     1 WORD(10)=44
      IF(WORD(12).EQ.23.AND.WORD(17).EQ.23)  WORD(12)=44
      IF(WORD(10).EQ.31.AND.WORD(12).EQ.44.AND.WORD(17).EQ.31)
     1  WORD(10)=44
      IF(WORD(12).EQ.31.AND.WORD(17).EQ.31)  WORD(12)=44
C
C ***                    COMPUTE VALUE OF L(F)  AND  R(F)
C
  361 WORD(1)=45
      WORD(19)=45
      IF(F.EQ.1) WORD(1)=46
      IF(F.EQ.2) WORD(19)=46
C
C ***                    LEFT JUSTIFY
C
      NC=0
      DO 37 L=1,19
      IF(WORD(L).EQ.44) GO TO 37
      NC=NC+1
      WORD(NC)=WORD(L)
   37 CONTINUE
      ND=NC+1
      IF(ND.GT.19) RETURN
      DO 38 L=ND,19
   38 WORD(L)=40
      RETURN
      END
```

Appendix

PART III

APPENDIX C

SAMPLE OUTPUT OF
SUBPROGRAM OUTPUT

```
    RESULTS OF REWRITE PASS NO.-  0

SYMBOL     M  F  K  B  C  L  Y  D  N  G  P  R  A  V  I  T  S     W      TY  RULE EL NE  REST SUR    X

   76      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0   0000      0     0  0  1    0    0    0
```

```
 RESULTS OF REWRITE PASS NO.-  1

SYMBOL    M  F  K  B  C  L  Y  D  N  G  P  R  A  V  I  T  S    W     TY  RULE EL NE  REST SUB    X

  74      1  0  0  0  1  0  0  0  0  0  0  0  0  0  1  1  0   0000    0     0  2  1    0    0    0
  92      1  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  0   0000    0     0  3  1    0    0    0
```

```
   RESULTS OF REWRITE PASS NO.-  2

SYMBOL   M  F  K  B  C  L  Y  D  N  G  P  R  A  V  I  T  S    W     TY  RULE EL NE  REST SUB   X

  64     1  1  0  0  3  0  0  0  1  1  3  0  0  1  1  1  0   0000    0    0   1  1    0    0    0
  92     1  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  1   .***    1    0   1  1    0    0    0
```

```
   RESULTS OF REWRITE PASS NO.-  3

SYMBOL     M  F  K  B  C  L  Y  D  N  G  P  R  A  V  I  T  S     W      TY   RULE EL NE  REST SUB    X

  62       1  0  0  0  1  0  0  0  1  1  3  0  0  1  1  1  0    0000     0      0  1  1    0    0     0
  92       1  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  1    .***     1      0  1  1    0    0     0
```

```
    RESULTS OF REWRITE PASS NO.-   4

SYMBOL    M  F  K  B  C  L  Y  D  N  G  P  R  A  V  I  T  S     W      TY   RULE EL NE  REST SUB    X

  42      1  2  0  0  1  0  0  2  1  1  3  0  0  0  0  0  0    0000     0     0  1  1     0    0    0
  53      1  1  0  0  1  0  0  0  1  1  3  0  7  1  1  1  0    0000     0     0  2  1     0    0    0
  92      1  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  1    .***     1     0  1  1     0    0    0
```

```
   RESULTS OF REWRITE PASS NO.-  5

SYMBOL:   M  F  K  B  C  L  Y  D  N  G  P  R  A  V  I  T  S     W      TY  RULE EL  NE  REST SUB    X

  26      1  2  0  0  1  0  0  2  1  1  3  0  0  0  0  0  0   0000     0     0   1   1    0    0    0
  36      1  0  0  0  1  0  0  0  1  1  3  0  7  1  1  1 12   SAL*     0     0   2   1    1   14    0
  50      1  1  0  0  7  0  0  0  0  0  0  0  7  0  0  0  0   0000     0     0   3   1    1   14    0
  92      1  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  1   .***     1     0   1   1    0    0    0
```

```
RESULTS OF REWRITE PASS NO.-  5

SYMBOL    M  F  K  B  C  L  Y  D  N  G  P  R  A  V  I  T  S    W      TY  RULE EL NE  REST SUB   X

   23     1  0  0  0  1  0  0  2  1  1  3  0  0  0  0  0  0   0000    0    0   1  1   0    0    0
   35     1  0  0  0  1  0  0  0  1  1  3  0  7  1  1  1 12   SAL*    0    0   1  1   2   10    0
   48     1  0  0  0  3  0  0  0  0  0  0  0  7  0  0  0  0   0000    0    0   1  1   0    0    0
   92     1  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  1   .***    1    0   1  1   0    0    0
```

3-C-7

```
RESULTS OF REWRITE PASS NO.-  7

SYMBOL    M  F  K  P  C  L  Y  D  N  G  P  R  A  V  I  T  S     W      TY  RULE EL NE  REST SUB    X

  22      1  0  0  0  1  0  0  2  1  1  3  0  0  0  0  0  0   0000     0     0  2  1    0    0    0
  34      1  0  0  0  3  0  0  0  1  1  3  0  7  1  1  1 12   SAL*     0     0  1  1   11   17    0
  43      1  1  0  0  2  0  0  2  0  0  0  0  0  0  0  0  0   0000     0     0  3  1    0    0    0
  30      1  4  0  0  3  0  0  0  0  0  0  0  0  0  0  0  0   0000     0     0  4  1    0    0    0
  72      1  1  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0   0000     0     0  5  1    0    0    0
  92      1  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  1   .***     1     0  1  1    0    0    0
```

```
RESULTS OF REWRITE PASS NO.-   8

SYMBOL    M  F  K  B  C  L  Y  D  N  G  P  R  A  V  I  T  S      W        TY  RULE EL NE  REST SUR    X


  6       1  2  0  0  2  0  0  2  1  1  3  0  0  0  0  0  0    0000       0     0  1  1    0    0     0
 33       1  0  0  0  1  0  0  0  1  1  3  0  7  1  1  1 12    SAL*       0     0  1  1   11   17     0
 39       1  1  0  0  1  0  0  2  0  0  0  0  0  0  0  0  0    0000       0     0  1  1    0    0     0
 27       1  1  0  0  1  1  0  0  0  0  0  0  0  0  0  0  0    0000       0     0  1  1    0    0     0
 92       1  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  0    0000       0     0  1  1    0    0     0
 92       1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0    0000       0     0  2  1    0    0     0
 76       1  1  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0    0000       0     0  3  1    0    0     0
 32       1  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  0    0000       0     0  4  1    0    0     0
 92       1  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  1    .***       1     0  1  1    0    0     0
```

```
   RESULTS OF REWRITE PASS NO.-   9

SYMBOL     M  F  K  B  C  L  Y  D  N  G  P  R  A  V  I  T  S      W       TY  RULE EL NE  REST SUB    X

  83       1  0  0  0  1  1  0  0  0  0  0  0  0  0  0  0  0     0000      0     0   1  1    0    0    0
  87       1  0  0  0  1  0  0  0  1  1  3  0  0  0  0  0  1     CB**      1     0   2  1    0    0    0
  32       1  0  0  0  1  0  0  0  1  1  3  0  7  1  1  1 12     SAL*      0     0   1  1    5   17    0
  88       1  0  0  0  1  1  0  0  0  0  0  0  0  0  0  0  0     0000      0     0   1  1    0    0    0
  26       1  0  0  0  1  0  0  2  1  1  3  0  0  0  0  0  0     0000      0     0   2  1    0    0    0
  80       1  0  0  0  3  1  0  0  0  0  0  0  0  0  0  0  1     POM*      1     0   1  1    0    0    0
  92       1  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  1     :***      1     0   1  1    0    0    0
  32       1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1     ''**      1     0   1  1    0    0    0
  75       1  1  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0     0000      0     0   2  1    0    0    0
  92       1  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  0     0000      0     0   3  1    0    0    0
  32       1  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  1     ''**      1     0   1  1    0    0    0
  92       1  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  1     .***      1     0   1  1    0    0    0
  RESULTS OF REWRITE PASS NO.- 10
SYMBOL    M  F  K  R  C  L  Y  D  N  G  P  R  A  V  I  T  S     W       TY  RULE EL  NE  REST SUB    X

  93      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1    H***     1     0   1   1    0    0    0
  87      1  0  0  0  1  0  0  0  1  1  3  0  0  0  0  0  1    CB**     1     0   2   1    0    0    0
  94      1  0  0  0  1  1  0  0  1  1  3  0  7  1  1  1 12    SAL*     0     0   1   1    0    0    0
  88      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 27    *AWT     1     0   1   1    0    0    0
  23      1  0  0  0  1  0  0  2  1  1  3  0  0  0  0  0  0    0000     0     0   1   1    0    0    0
  80      1  0  0  0  3  1  0  0  0  0  0  0  0  0  0  0  1    POM*     1     0   1   1    0    0    0
  92      1  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  1    :***     1     0   1   1    0    0    0
  92      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1    ''**     1     0   1   1    0    0    0
  71      1  0  0  0  1  0  0  0  1  1  2  0  0  0  0  0  0    0000     0     0   1   1    0    0    0
  92      1  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  1    ?***     1     0   1   1    0    0    0
  92      1  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  1    ''**     1     0   1   1    0    0    0
  92      1  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  1    .***     1     0   1   1    0    0    0
```

```
RESULTS OF REWRITE PASS NO.- 11

SYMBOL   M  F  K  B  C  L  Y  D  N  G  P  R  A  V  I  T  S     W      TY  RULE EL  NE  REST SUB   X

  83     1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   H***     1     0  1   1    0    0    0
  87     1  0  0  0  1  0  0  0  1  1  3  0  0  0  0  0  1   CB**     1     0  2   1    0    0    0
  94     1  0  0  0  1  0  0  0  1  1  3  0  7  1  1  1 12   SAL*     1     0  1   1    0    0    0
  88     1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 27   *AWT     1     0  1   1    0    0    0
  22     1  0  0  0  1  0  0  2  1  1  3  0  0  0  0  0  0   0000     0     0  2   1    0    0    0
  90     1  0  0  0  3  1  0  0  0  0  0  0  0  0  0  0  1   POM*     1     0  1   1    0    0    0
  92     1  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  1   :***     1     0  1   1    0    0    0
  92     1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   ''**     1     0  1   1    0    0    0
  90     1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  0   0000     0     0  1   1    0    0    0
  64     1  1  0  0  3  0  0  0  1  1  2  0  0  1  1  3  0   0000     0     0  2   1    0    0    0
  92     1  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  1   ?***     1     0  1   1    0    0    0
  92     1  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  1   ''**     1     0  1   1    0    0    0
  92     1  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  1   .***     1     0  1   1    0    0    0
```

```
RESULTS OF REWRITE PASS NO.- 12

SYMBOL     M  F  K  B  C  L  Y  D  N  G  P  R  A  V  I  T  S     W       TY  RULE EL NE  REST SUR    X

   63      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1    H***     1     0  1  1    0    0    0
   37      1  0  0  0  1  0  0  0  1  1  3  0  0  0  0  0  1    CB**     1     0  2  1    0    0    0
   94      1  0  0  0  1  0  0  0  1  1  3  0  7  1  1  1 12    SAL*     1     0  1  1    0    0    0
   98      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 27    *AWT     1     0  1  1    0    0    0
    6      1  2  0  0  7  0  0  2  1  1  3  0  0  0  0  0  0    0000     0     0  1  1    0    0    0
   80      1  0  0  0  3  1  0  0  0  0  0  0  0  0  0  0  1    POM*     1     0  1  1    0    0    0
   92      1  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  1    :***     1     0  1  1    0    0    0
   92      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1    ''**     1     0  1  1    0    0    0
   90      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1    H***     1     0  1  1    0    0    0
   62      1  0  0  0  1  0  0  0  1  1  2  0  0  1  1  3  0    0000     0     0  1  1    0    0    0
   92      1  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  1    ?***     1     0  1  1    0    0    0
   92      1  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  1    ''**     1     0  1  1    0    0    0
   92      1  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  1    .***     1     0  1  1    0    0    0
```

```
 RESULTS OF REWRITE PASS NO:- 13

SYMBOL   W  F  K  R  C  L  Y  D  N  G  P  R  A  V  I  T  S     W      TY  RULE EL NE  REST SUB    X

  83     1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   H***     1     0  1  1    0    0     0
  87     1  0  0  0  1  0  0  0  1  1  3  0  0  0  0  0  1   CB**     1     0  2  1    0    0     0
  94     1  0  0  0  1  0  0  0  1  1  3  0  7  1  1  1 12   SAL*     1     0  1  1    0    0     0
  88     1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 27   *AWT     1     0  1  1    0    0     0
  87     1  4  0  0  2  1  0  0  1  1  3  0  0  0  0  0  1   AWRY     1     0  1  1    1    6     0
  90     1  0  0  0  3  1  0  0  0  0  0  0  0  0  0 .0  1   POM*     1     0  1  1    0    0     0
  92     1  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  1   :***     1     0  1  1    0    0     0
  92     1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   ''**     1     0  1  1    0    0     0
  90     1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   H***     1     0  1  1    0    0     0
  42     1  2  0  0  2  0  0  0  1  1  2  0  0  0  0  0  0   0000     0     0  1  1    0    0     0
  53     1  1  0  0  1  0  0  0  1  1  2  0  6  1  1  3  0   0000     0     0  2  1    0    0     0
  92     1  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  1   ?***     1     0  1  1    0    0     0
  92     1  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  1   ''**     1     0  1  1    0    0     0
  92     1  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  1   .***     1     0  1  1    0    0     0
 RESULTS OF REWRITE PASS NO.- 14

SYMBOL     M  F  K  B  C  L  Y  D  N  S  P  R  A  V  I  T  S     W       TY  RULE EL NE  REST SUB    X


  83       1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   H***      1     0  1  1    0    0    0
  87       1  0  0  0  1  0  0  0  1  1  3  0  0  0  0  0  1   CB**      1     0  2  1    0    0    0
  94       1  0  0  0  1  0  0  0  1  1  3  0  7  1  1  1 12   SAL*      1     0  1  1    0    0    0
  86       1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 27   *AWT      1     0  1  1    0    0    0
  87       1  4  0  0  2  1  0  0  1  1  3  0  0  0  0  0  1   AWRY      1     0  1  1    1    6    0
  86       1  0  0  0  3  1  0  0  0  0  0  0  0  0  0  0  1   POM*      1     0  1  1    0    0    0
  92       1  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  1   :***      1     0  1  1    0    0    0
  92       1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   ''**      1     0  1  1    0    0    0
  90       1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   H***      1     0  1  1    0    0    0
  41       1  2  0  0  1  0  0  2  1  1  2  0  0  0  0  0  0   0000      0     0  1  1    0    0    0
  36       1  0  0  0  1  0  0  0  1  1  2  0  6  1  1  3 12   R&H*      0     0  2  1    1   14    0
  50       1  1  0  0  6  0  0  0  0  0  0  0  6  0  0  0  0   0000      0     0  3  1    1   14    0
  92       1  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  1   ?***      1     0  1  1    0    0    0
  92       1  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  1   ''**      1     0  1  1    0    0    0
  92       1  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  1   .***      1     0  1  1    0    0    0
```

```
RESULTS OF REWRITE PASS NO.- 15

SYMBOL    M  F  K  B  C  L  Y  D  N  G  P  R  A  V  I  T  S     W      TY  RULE EL NE  REST SUB   X

  83      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1    H***    1     0   1  1    0    0    0
  87      1  0  0  0  1  0  0  0  1  1  3  0  0  0  0  0  1    CB**    1     0   2  1    0    0    0
  94      1  0  0  0  1  0  0  0  1  1  3  0  7  1  1  1 12    SAL*    1     0   1  1    0    0    0
  98      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 27    *AWT    1     0   1  1    0    0    0
  37      1  4  0  0  2  1  0  0  1  1  3  0  0  0  0  0  1    AWRY    1     0   1  1    1    6    0
  80      1  0  0  0  3  1  0  0  0  0  0  0  0  0  0  0  1    POM*    1     0   1  1    0    0    0
  92      1  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  1    :***    1     0   1  1    0    0    0
  92      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1    ''**    1     0   1  1    0    0    0
  90      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1    H***    1     0   1  1    0    0    0
  91      1  0  0  0  2  0  0  0  1  1  2  0  0  0  0  0  0    0000    1     0   1  1    0    0    0
  35      1  0  0  0  1  0  0  0  1  1  2  0  6  1  1  3 12    R&H*    0     0   1  1    2   10    0
  99      1  0  0  0  1  0  0  0  0  0  0  1  0  0  0  0  0    0000    0     0   2  1    0    0    0
  56      1  0  0  0  1  0  0  0  0  0  0  0  3  0  0  0  0    0000    0     0   3  1    0    0    0
  92      1  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  1    ?***    1     0   1  1    0    0    0
  92      1  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  1    ''**    1     0   1  1    0    0    0
  92      1  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  1    .***    1     0   1  1    0    0    0
```

```
RESULTS OF REWRITE PASS NO.- 16

SYMBOL   M  F  K  B  C  L  Y  D  N  G  P  R  A  V  I  T  S     W      TY  RULE EL NE  REST SUB   X

  83     1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1    H***     1    0   1  1    0    0    0
  87     1  0  0  0  1  0  0  0  1  1  3  0  0  0  0  0  1    CB**     1    0   2  1    0    0    0
  94     1  0  0  0  1  0  0  0  1  1  3  0  7  1  1  1 12    SAL*     1    0   1  1    0    0    0
  88     1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 27    *AWT     1    0   1  1    0    0    0
  87     1  4  0  0  2  1  0  0  1  1  3  0  0  0  0  0  1    AWRY     1    0   1  1    1    6    0
  90     1  0  0  0  3  1  0  0  0  0  0  0  0  0  0  0  1    POM*     1    0   1  1    0    0    0
  92     1  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  1    :***     1    0   1  1    0    0    0
  92     1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1    ''**     1    0   1  1    0    0    0
  90     1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1    H***     1    0   1  1    0    0    0
  91     1  0  0  0  2  0  0  0  1  1  2  0  0  0  0  0  0    0000     1    0   1  1    0    0    0
  34     1  0  0  0  3  0  0  0  1  1  2  0  6  1  1  3 12    R8H*     0    0   1  1   11   17    0
  89     1  0  0  0  1  0  0  0  0  0  0  1  0  0  0  0  1    L***     1    0   1  1    0    0    0
  96     1  0  0  0  3  3  0  0  0  0  0  0  3  0  0  0 14    RAH*     1    0   1  1    0    0    0
  50     1  1  0  0  3  0  0  0  0  0  0  0  3  0  0  0  0    0000     0    0   2  1    0    0    0
  92     1  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  1    ?***     1    0   1  1    0    0    0
  92     1  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  1    ''**     1    0   1  1    0    0    0
  92     1  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  1    .***     1    0   1  1    0    0    0
```

```
 RESULTS OF REWRITE PASS NO.- 17

SYMBOL     M  F  K  B  C  L  Y  D  N  G  P  R  A  V  I  T  S     W       TY  RULE EL NE  REST SUB    X


  83       1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   H***      1      0  1  1    0    0    0
  87       1  0  0  0  1  0  0  0  1  1  3  0  0  0  0  0  1   CB**      1      0  2  1    0    0    0
  94       1  0  0  0  1  0  0  0  1  1  3  0  7  1  1  1 12   SAL*      1      0  1  1    0    0    0
  88       1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 27   *AWT      1      0  1  1    0    0    0
  87       1  4  0  0  2  1  0  0  1  1  3  0  0  0  0  0  1   AWRY      1      0  1  1    1    6    0
  80       1  0  0  0  3  1  0  0  0  0  0  0  0  0  0  0  1   POM*      1      0  1  1    0    0    0
  92       1  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  1   :***      1      0  1  1    0    0    0
  92       1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   ''**      1      0  1  1    0    0    0
  90       1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   H***      1      0  1  1    0    0    0
  91       1  0  0  0  2  0  0  0  1  1  2  0  0  0  0  0  0   0000      1      0  1  1    0    0    0
  33       1  0  0  0  1  0  0  0  1  1  2  0  6  1  1  3 12   R8H*      0      0  1  1   11   17    0
  89       1  0  0  0  1  0  0  0  0  0  0  1  0  0  0  0  1   L***      1      0  1  1    0    0    0
  96       1  0  0  0  3  3  0  0  0  0  0  0  3  0  0  0 14   RAH*      1      0  1  1    0    0    0
  46       1  0  0  0  3  0  0  0  0  0  0  0  3  0  0  0  0   0000      0      0  1  1    0    0    0
  92       1  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  1   ?***      1      0  1  1    0    0    0
  92       1  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  1   ''**      1      0  1  1    0    0    0
  92       1  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  1   .***      1      0  1  1    0    0    0
```

```
RESULTS OF REWRITE PASS NO.- 13

SYMBOL    M  F  K  B  C  L  Y  D  N  G  P  R  A  V  I  T  S     W      TY  RULE EL NE  REST SUB   X

  83      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1    H***    1     0   1  1    0    0    0
  87      1  0  0  0  1  0  0  0  1  1  3  0  0  0  0  0  1    CB**    1     0   2  1    0    0    0
  94      1  0  0  0  1  0  0  0  1  1  3  0  7  1  1  1 12    SAL*    1     0   1  1    0    0    0
  88      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 27    *AWT    1     0   1  1    0    0    0
  87      1  4  0  0  2  1  0  0  1  1  3  0  0  0  0  0  1    AWRY    1     0   1  1    1    6    0
  80      1  0  0  0  3  1  0  0  0  0  0  0  0  0  0  0  1    POM*    1     0   1  1    0    0    0
  92      1  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  1    :***    1     0   1  1    0    0    0
  92      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1    ''**    1     0   1  1    0    0    0
  90      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1    H***    1     0   1  1    0    0    0
  91      1  0  0  0  2  0  0  0  1  1  2  0  0  0  0  0  0    0000    1     0   1  1    0    0    0
  31      1  0  0  0  1  0  0  1  1  1  2  0  6  1  0  0 12    R8H*    0     0   1  1    0    0    0
  89      1  0  0  0  1  0  0  0  0  0  0  1  0  0  0  0  1    L***    1     0   1  1    0    0    0
  96      1  0  0  0  3  3  0  0  0  0  0  0  3  0  0  0 14    RAH*    1     0   1  1    0    0    0
  43      1  1  0  0  2  0  0  2  0  0  0  0  0  0  0  0  0    0000    0     0   3  1    0    0    0
  92      1  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  1    ?***    1     0   1  1    0    0    0
  92      1  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  1    ''**    1     0   1  1    0    0    0
  92      1  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  1    .***    1     0   1  1    0    0    0
```

```
RESULTS OF REWRITE PASS NO.- 19

SYMBOL   M  F  K  P  C  L  Y  D  N  G  P  R  A  V  I  T  S     W      TY  RULE EL NE  REST SUB   X

  83     1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   H***     1     0   1  1    0    0    0
  87     1  0  0  0  1  0  0  0  1  1  3  0  0  0  0  0  1   CB**     1     0   2  1    0    0    0
  94     1  0  0  0  1  0  0  0  1  1  3  0  7  1  1  1 12   SAL*     1     0   1  1    0    0    0
  98     1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 27   *AWT     1     0   1  1    0    0    0
  87     1  4  0  0  2  1  0  0  1  1  3  0  0  0  0  0  1   AWRY     1     0   1  1    1    6    0
  80     1  0  0  0  3  1  0  0  0  0  0  0  0  0  0  0  1   POM*     1     0   1  1    0    0    0
  92     1  0  0  0  4  0  0  0  0  0  0  0  0  0  0 .0  1   :***     1     0   1  1    0    0    0
  92     1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   ''**     1     0   1  1    0    0    0
  90     1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   H***     1     0   1  1    0    0    0
  91     1  0  0  0  2  0  0  0  1  1  2  0  0  0  0  0  0   0000     1     0   1  1    0    0    0
  81     1  0  0  0  1  1  0  0  1  1  2  0  6  1  0  0 12   RBH*     0     0   1  1    2    9    0
  89     1  0  0  0  1  0  0  0  0  0  0  1  0  0  0  0  1   L***     1     0   1  1    0    0    0
  96     1  0  0  0  3  3  0  0  0  0  0  0  3  0  0  0 14   RAH*     1     0   1  1    0    0    0
  39     1  1  0  0  1  0  0  2  0  0  0  0  0  0  0  0  0   0000     0     0   1  1    0    0    0
  92     1  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  1   ?***     1     0   1  1    0    0    0
  92     1  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  1   ''**     1     0   1  1    0    0    0
  92     1  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  1   .***     1     0   1  1    0    0    0
```

219

```
RESULTS OF REWRITE PASS NO.- 20

SYMBOL   M F K B C L Y D N G P R A V I T  S    W     TY  RULE EL NE  REST SUB  X

  83     1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0  1   H***   1     0   1  1   0    0   0
  87     1 0 0 0 1 0 0 0 1 1 3 0 0 0 0 0  1   CB**   1     0   2  1   0    0   0
  94     1 0 0 0 1 0 0 0 1 1 3 0 7 1 1 1 12   SAL*   1     0   1  1   0    0   0
  88     1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 27   *AWT   1     0   1  1   0    0   0
  87     1 4 0 0 2 1 0 0 1 1 3 0 0 0 0 0  1   AWRY   1     0   1  1   1    6   0
  80     1 0 0 0 3 1 0 0 0 0 0 0 0 0 0 0  1   POM*   1     0   1  1   0    0   0
  92     1 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0  1   :***   1     0   1  1   0    0   0
  92     1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0  1   ''**   1     0   1  1   0    0   0
  90     1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0  1   H***   1     0   1  1   0    0   0
  91     1 0 0 0 2 0 0 0 1 1 2 0 0 0 0 0  0   OOCO   1     0   1  1   0    0   0
  81     1 0 0 0 1 0 0 0 1 1 2 0 6 1 0 0 12   R&H*   1     0   1  1   0    0   0
  89     1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0  1   L***   1     0   1  1   0    0   0
  96     1 0 0 0 3 3 0 0 0 0 0 0 3 0 0 0 14   RAH*   1     0   1  1   0    0   0
  88     1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0  0   OOCO   0     0   1  1   0    0   0
  26     1 0 0 0 1 0 0 2 1 1 3 0 0 0 0 0  0   OOOO   0     0   2  1   0    0   0
  92     1 0 0 0 5 0 0 0 0 0 0 0 0 0 0 0  1   ?***   1     0   1  1   0    0   0
  92     1 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0  1   ''**   1     0   1  1   0    0   0
  92     1 0 0 0 6 0 0 0 0 0 0 0 0 0 0 0  1   .***   1     0   1  1   0    0   0
```

```
RESULTS OF REWRITE PASS NO.- 21
SYMBOL     M  F  K  B  C  L  Y  D  N  G  P  R  A  V  I  T  S     W       TY  RULE EL NE  REST SUB   X

  83       1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   H***      1     0  1  1    0    0    0
  87       1  0  0  0  1  0  0  0  1  1  3  0  0  0  0  0  1   CB**      1     0  2  1    0    0    0
  94       1  0  0  0  1  0  0  0  1  1  3  0  7  1  1  1 12   SAL*      1     0  1  1    0    0    0
  88       1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 27   *AWT      1     0  1  1    0    0    0
  87       1  4  0  0  2  1  0  0  1  1  3  0  0  0  0  0  1   AWRY      1     0  1  1    1    6    0
  80       1  0  0  0  3  1  0  0  0  0  0  0  0  0  0  0  1   POM*      1     0  1  1    0    0    0
  92       1  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  1   :***      1     0  1  1    0    0    0
  92       1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   ''**      1     0  1  1    0    0    0
  90       1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   H***      1     0  1  1    0    0    0
  91       1  0  0  0  2  0  0  0  1  1  2  0  0  0  0  0  0   0000      1     0  1  1    0    0    0
  81       1  0  0  0  1  0  0  0  1  1  2  0  6  1  0  0 12   R8H*      1     0  1  1    0    0    0
  89       1  0  0  0  1  0  0  0  0  0  0  1  0  0  0  0  1   L***      1     0  1  1    0    0    0
  96       1  0  0  0  3  3  0  0  0  0  0  0  3  0  0  0 14   RAH*      1     0  1  1    0    0    0
  88       1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 27   *AWT      1     0  1  1    0    0    0
  23       1  0  0  0  2  0  0  2  1  1  3  0  0  0  0  0  0   0000      0     0  1  1    0    0    0
  92       1  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  1   ?***      1     0  1  1    0    0    0
  92       1  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  1   ''**      1     0  1  1    0    0    0
  92       1  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  1   .***      1     0  1  1    0    0    0
```

```
RESULTS OF REWRITE PASS NO.- 22

SYMBOL    M  F  K  B  C  L  Y  D  N  G  P  R  A  V  I  T  S     W      TY  RULE EL NE  REST SUB   X

  93      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1    H***    1     0   1  1    0    0    0
  87      1  0  0  0  1  0  0  0  1  1  3  0  0  0  0  0  1    CB**    1     0   2  1    0    0    0
  94      1  0  0  0  1  0  0  0  1  1  3  0  7  1  1  1 12    SAL*    1     0   1  1    0    0    0
  98      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 27    *AWT    1     0   1  1    0    0    0
  87      1  4  0  0  2  1  0  0  1  1  3  0  0  0  0  0  1    AWRY    1     0   1  1    1    6    0
  80      1  0  0  0  3  1  0  0  0  0  0  0  0  0  0  0  1    POM*    1     0   1  1    0    0    0
  92      1  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  1    :***    1     0   1  1    0    0    0
  92      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1    ''**    1     0   1  1    0    0    0
  90      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1    H***    1     0   1  1    0    0    0
  91      1  0  0  0  2  0  0  0  1  1  2  0  0  0  0  0  0    0000    1     0   1  1    0    0    0
  81      1  0  0  0  1  0  0  0  1  1  2  0  6  1  0  0 12    R&H*    1     0   1  1    0    0    0
  89      1  0  0  0  1  0  0  0  0  0  0  1  0  0  0  0  1    L***    1     0   1  1    0    0    0
  96      1  0  0  0  3  3  0  0  0  0  0  0  3  0  0  0 14    RAH*    1     0   1  1    0    0    0
  88      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 27    *AWT    1     0   1  1    0    0    0
  85      1  0  0  0  1  0  0  0  1  1  3  0  0  0  0  0 29    3YYT    1     0   2  1    0    0    0
  22      1  0  0  0  1  0  0  2  1  1  3  0  0  0  0  0  0    0000    0     0   4  1    0    0    0
  92      1  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  1    ?***    1     0   1  1    0    0    0
  92      1  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  1    ''**    1     0   1  1    0    0    0
  92      1  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  1    .***    1     0   1  1    0    0    0
```

```
RESULTS OF REWRITE PASS NO.- 23

SYMBOL   M  F  K  R  C  L  Y  D  N  G  P  R  A  V  I  T  S    W     TY  RLLE EL NE  REST SUR   X


 93      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   H***    1    0   1  1    0    0    0
 87      1  0  0  0  1  0  0  0  1  1  3  0  0  0  0  0  1   CB**    1    0   2  1    0    0    0
 94      1  0  0  0  1  0  0  0  1  1  3  0  7  1  1  1 12   SAL*    1    0   1  1    0    0    0
 88      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 27   *AWT    1    0   1  1    0    0    0
 87      1  4  0  0  2  1  0  0  1  1  3  0  0  0  0  0  1   AWRY    1    0   1  1    1    6    0
 80      1  0  0  0  3  1  0  0  0  0  0  0  0  0  0  0  1   POM*    1    0   1  1    0    0    0
 92      1  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  1   :***    1    0   1  1    0    0    0
 92      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   ''**    1    0   1  1    0    0    0
 90      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   H***    1    0   1  1    0    0    0
 91      1  0  0  0  2  0  0  0  1  1  2  0  0  0  0  0  0   0000    1    0   1  1    0    0    0
 81      1  0  0  0  1  0  0  0  1  1  2  0  6  1  0  0 12   RBH*    1    0   1  1    0    0    0
 89      1  0  0  0  1  0  0  0  0  0  0  1  0  0  0  0  1   L***    1    0   1  1    0    0    0
 96      1  0  0  0  7  3  0  0  0  0  0  0  3  0  0  0 14   RAH*    1    0   1  1    0    0    0
 88      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 27   *AWT    1    0   1  1    0    0    0
 85      1  0  0  0  1  0  0  0  1  1  3  0  0  0  0  0 29   BYYT    1    0   2  1    0    0    0
  6      1  2  0  0  2  0  0  2  1  1  3  0  0  0  0  0  0   0000    0    0   1  1    0    0    0
 92      1  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  1   ?***    1    0   1  1    0    0    0
 92      1  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  1   ''**    1    0   1  1    0    0    0
 92      1  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  1   .***    1    0   1  1    0    0    0
```

```
RESULTS OF REWRITE PASS NO.- 24

SYMBOL    M  F  K  B  C  L  Y  D  N  G  P  R  A  V  I  T  S    W     TY  RULE EL NE  REST SUB   X

  83      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   H***   1      0  1  1    0    0    0
  87      1  0  0  0  1  0  0  0  1  1  3  0  0  0  0  0  1   CB**   1      0  2  1    0    0    0
  94      1  0  0  0  1  0  0  0  1  1  3  0  7  1  1  1 12   SAL*   1      0  1  1    0    0    0
  88      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 27   *AWT   1      0  1  1    0    0    0
  87      1  4  0  0  2  1  0  0  1  1  3  0  0  0  0  0  1   AWRY   1      0  1  1    1    6    0
  80      1  0  0  0  3  1  0  0  0  0  0  0  0  0  0  0  1   POM*   1      0  1  1    0    0    0
  92      1  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  1   :***   1      0  1  1    0    0    0
  92      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   ''**   1      0  1  1    0    0    0
  90      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1   H***   1      0  1  1    0    0    0
  91      1  0  0  0  2  0  0  0  1  1  2  0  0  0  0  0  0   0000   1      0  1  1    0    0    0
  81      1  0  0  0  1  0  0  0  1  1  2  0  6  1  0  0 12   R&H*   1      0  1  1    0    0    0
  89      1  0  0  0  1  0  0  0  0  0  0  1  0  0  0  0  1   I***   1      0  1  1    0    0    0
  96      1  0  0  0  3  3  0  0  0  0  0  0  3  0  0  0 14   RAH*   1      0  1  1    0    0    0
  88      1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 27   *AWT   1      0  1  1    0    0    0
  85      1  0  0  0  1  0  0  0  1  1  3  0  0  0  0  0 29   BYYT   1      0  2  1    0    0    0
  83      1  0  0  0  1  1  0  0  0  0  0  0  0  0  0  0  0   0000   0      0  1  1    0    0    0
  87      1  0  0  0  1  0  0  0  1  1  3  0  0  0  0  0  1   KNC*   1      1  2  1    0    0    1
  92      1  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  1   ?***   1      0  1  1    0    0    0
  92      1  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  1   ''**   1      0  1  1    0    0    0
  92      1  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  1   .***   1      0  1  1    0    0    0
```

```
RESULTS OF REWRITE PASS NO.- 25

SYMBOL     M  F  K  R  C  L  Y  D  N  G  P  R  A  V  I  T  S     W      TY  RULE EL WE  REST SUB   X


  83       1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1    H***    1     0   1  1    0    0    0
  97       1  0  0  0  1  0  0  0  1  1  3  0  0  0  0  0  1    CB**    1     0   2  1    0    0    0
  94       1  0  0  0  1  0  0  0  1  1  3  0  7  1  1  1 12    SAL*    1     0   1  1    0    0    0
  88       1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 27    *AWT    1     0   1  1    0    0    0
  87       1  4  0  0  2  1  0  0  1  1  3  0  0  0  0  0  1    AWRY    1     0   1  1    1    6    0
  90       1  0  0  0  3  1  0  0  0  0  0  0  0  0  0  0  1    POM*    1     0   1  1    0    0    0
  92       1  0  0  0  4  0  0  0  0  0  0  0  0  0  0  0  1    :***    1     0   1  1    0    0    0
  92       1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1    ''**    1     0   1  1    0    0    0
  90       1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1    H***    1     0   1  1    0    0    0
  91       1  0  0  0  2  0  0  0  1  1  2  0  0  0  0  0  0    DDCO    1     0   1  1    0    0    0
  91       1  0  0  0  1  0  0  0  1  1  2  0  6  1  0  0 12    R&H*    1     0   1  1    0    0    0
  89       1  0  0  0  1  0  0  0  0  0  0  1  0  0  0  0  1    L***    1     0   1  1    0    0    0
  96       1  0  0  0  3  3  0  0  0  0  0  0  3  0  0  0 14    RAH*    1     0   1  1    0    0    0
  88       1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0 27    *AWT    1     0   1  1    0    0    0
  95       1  0  0  0  1  0  0  0  1  1  3  0  0  0  0  0 29    BYYT    1     0   2  1    0    0    0
  83       1  0  0  0  1  0  0  0  0  0  0  0  0  0  0  0  1    H***    1     0   1  1    0    0    0
  97       1  0  0  0  1  0  0  0  1  1  3  0  0  0  0  0  1    KNC*    1     1   2  1    0    0    1
  92       1  0  0  0  5  0  0  0  0  0  0  0  0  0  0  0  1    ?***    1     0   1  1    0    0    0
  92       1  0  0  0  2  0  0  0  0  0  0  0  0  0  0  0  1    ''**    1     0   1  1    0    0    0
  92       1  0  0  0  6  0  0  0  0  0  0  0  0  0  0  0  1    .***    1     0   1  1    0    0    0
```

Appendix

PART III

APPENDIX D

LIST OF SENTENCE DESCRIPTION DATA CARD TYPES

PART III

APPENDIX D

This appendix contains the list of sentence description input data card types. These cards are used to define sentences to be generated by the computer. See Section 3.3.2.3 for instructions on the use of these data card types.

Z

CARD TYPE: 1

SENTENCE CONSTITUENT: Prepositional Pronoun Phrase

VARIABLES: M,K,B,C,Y,R

C=	OPTION	ADD CARD TYPES TO LIST:
-	M = o, constituent not to be used	none
1	For all cases where M=1	89A(R=R) 91 (C=3)
Note 1	If specified by the governing node, remove this card after coding.	

CARD FORMAT:

		1		M	K B 1	Y		R			
1	2 3 4 5	6	7 8 9 10	11 12 13 14 15	16 17 18 19 20	21 22 23 24 25	26 27 28 29 30	31 32 33 34 35	36 37 38 39 40	41 42 43 44 45	46 47 48 49 50

CARD TYPE: 2 Vqo

SENTENCE CONSTITUENT: Objective Interogative Verb Phrase

VARIABLES: K,B,R,A,Y,I,T

C=	OPTION	ADD CARD TYPES TO LIST:
A=3		*53(R=R,A=2,K=B=Y=V=I =T=0)
A=4		*53(R=R,A=5,K=B=Y=V =I=T=0)
A=8		*53(R=R,A=3,K=B=Y=V =I=T=0)
A=other	logical error	
		* NOTE 1: Cord type 53 should be removed from list after coding.

CARD FORMAT:

					2										K		B		1														R		A		V		I		T								
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 3 Apa

SENTENCE CONSTITUENT: Basic P~~a~~ost-Nominal Adjective Phrase

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Non-comparative degree	77 80(C=6, Y=0)
2	Comparative degree	77 38(m=1,y=0,R=4,5)
3	Superlative degree (associated Basic Noun Phrase, Npa, should be definite C=2)	77 38 (M=1, Y=0, R=2)

CARD FORMAT:

					3														C																													
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49

222

CARD TYPE: 4 Ap

SENTENCE CONSTITUENT: Post-Nominal Adjective Phrase

VARIABLES: M,K,B,C,Y

C=	OPTION	ADD CARD TYPES TO LIST:
-	M=0, phrase not used	none
1	for all	3

CARD FORMAT:

					4						M				K		B		1				Y																										
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

As

CARD TYPE: 5

SENTENCE CONSTITUENT: Adjectival Possessive Phrase

VARIABLES: M,K,B,C

C=	OPTION	ADD CARD TYPES TO LIST:
-	M=0 Phrase not used	None
1	Thing possessed is pronoun	91(C=3)
2	Thing possessed is noun	26(K=0, Y=0)

CARD FORMAT:

5 M K B C

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50

CARD TYPE: 6 N_a

SENTENCE CONSTITUENT: **Basic Noun Phrase**

VARIABLES: K,B,C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Noun is indefinite	87(C=1) K=B=0
2	Noun is definite, but not a proper noun	87(C=1) K=B≠0
3	Noun is proper noun or string of proper nouns	87(C ≠ 1)

CARD FORMAT:

					6										K		B		C																													
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49

CARD TYPE: 7 Sqo

SENTENCE CONSTITUENT: Objective Interogative Phrase

VARIABLES: C,Y,N,G,P

C=	OPTION	ADD CARD TYPES TO LIST:
1	Subject of verbal is not named	2(Y=N=G=P=0) R=R
2	Subject of verbal is named by noun or pronoun	42(N=G=P=0) 2(Y=N=G=P=0) R=R

CARD FORMAT:

					7														C				Y				N		G		P		R																
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 8* Ns

SENTENCE CONSTITUENT: Possessive Pronoun Noun Phrase

VARIABLES:

C=	OPTION	ADD CARD TYPES TO LIST:
1	For all cases	85(M=1, K=0) 91(C=3)

*After coding is complete, remove card from list

CARD FORMAT:

					8														1																														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 9 Rd

SENTENCE CONSTITUENT: Basic Demonstrative Pronoun Phrase

VARIABLES: M,C

C=	OPTION	ADD CARD TYPES TO LIST:
-	M=0, Demonstrative not to be used	none
1	Used in place of a noun in constructions like "this is the boy".	None
2	"This" used to modify a noun to specify nearness in place, time , or thought	None
3	"That" used to modify a noun to specify a relationship remote in place, time or thought	None

CARD FORMAT:

		9		M	C						
1	2 3 4 5.	6	7 8 9 10	11 12 13 14 15	16 17 18 19 20	21 22 23 24 25	26 27 28 29 30	31 32 33 34 35	36 37 38 39 40	41 42 43 44 45	46 47 48

CARD TYPE: 10 Ro

SENTENCE CONSTITUENT: Direct-Object Pronoun Phrase

VARIABLES: K,B,C,Y,N,G,P

C=	OPTION	ADD CARD TYPES TO LIST:
1	For all cases	None

CARD FORMAT:

				1	0										K		B		1				Y				N		G	P																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

Baa

CARD TYPE: 11

SENTENCE CONSTITUENT: Units Number Phrase

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Number 1	78A(C=1)
2	Number 2	78A(C=2)
3	Number 3	78($\overset{C=3}{S=1}$, W=SLS*)
	Number 4	78($\overset{C=3}{S=1}$, W=AREO)
	Number 5	78($\overset{C=3}{S=1}$, W=XMS*)
	Number 6	78($\overset{C=3}{S=1}$, W=SS**)
	Number 7	78($\overset{C=3}{S=1}$, W=SBO*)
	Number 8	78($\overset{C=3}{S=21}$, W=MNWH)
	Number 9	78($\overset{C=3}{S=1}$, W=TSO*)

CARD FORMAT:

				1	1														C																														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 12 B_{aa}

SENTENCE CONSTITUENT: **Tens Number Phrase**

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Number 10	None

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
				1	2														1																													

Bac

CARD TYPE: 13

SENTENCE CONSTITUENT: **Teens Number Phrase**

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Number 11	78A(C=1)
2	Number 12	78A(C=2)
3	Number 13	78($\overset{C=3}{S=1}$, W=SLS*)
	Number 14	78($\overset{C=3}{S=1}$, W=ARBO)
	Number 15	78($\overset{C=3}{S=1}$, W=XMS*)
	Number 16	78($\overset{C=3}{S=1}$, W=SS**)
	Number 17	78($\overset{C=3}{S=21}$, W=MNWH)
	Number 18	78($\overset{C=3}{S=21}$, W=MNWH)
	Number 19	78($\overset{C=3}{S=1}$, W=TSO*)

CARD FORMAT:

	1	3			C						
1	2 3 4 5	6	7 8 9 10	11 12 13 14 15	16 17 18 19 20	21 22 23 24 25	26 27 28 29 30	31 32 33 34 35	36 37 38 39 40	41 42 43 44 45	46 47 48 49 5

CARD TYPE: 14 **Bad**

SENTENCE CONSTITUENT: **Multi-Tens Number Phrase**

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Number 20	None
2	Number 30	78($\overset{C=3}{S=1}$, W=SLS*)
	Number 40	78($\overset{C=3}{S=1}$, W=ARBO)
	Number 50	78($\overset{C=3}{S=1}$, W=XMS*)
	Number 60	78($\overset{C=3}{S=1}$, W=SS**)
	Number 70	78($\overset{C=3}{S=1}$, W=SBO*)
	Number 80	78($\overset{C=3}{S=21}$, W=MNWH)
	Number 90	78($\overset{C=3}{S=1}$, W=TSO*)

CARD FORMAT:

				1	4														C																														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

Bae

CARD TYPE: 15

SENTENCE CONSTITUENT: Hundreds Number Phrase

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Number 100	84A(C=1)
2	Number 200	84A(C=2)
3	Number 300	84($\overset{C=3}{S=1}$, W=SLS*)
	Number 400	84($\overset{C=3}{S=1}$, W=ARBO)
	Number 500	84($\overset{C=3}{S=1}$, W=XMS*)
	Number 600	84($\overset{C=3}{S=1}$, W=SS**)
	Number 700	84($\overset{C=3}{S=1}$, W=SBO*)
	Number 800	84($\overset{C=3}{S=21}$, W=MNWH)
	Number 900	84($\overset{C=3}{S=1}$, W=TSO*)

CARD FORMAT:

				1	5														C																													
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49

CARD TYPE: 16

SENTENCE CONSTITUENT: Thousands Number Phrase

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Number 1,000	84(C=1)
2	Number 2,000	84(C=2)
3	Number 3,000	84($\overset{C=3}{S=1}$, W=SLS*)
	Number 4,000	84($\overset{C=3}{S=1}$, W=ARBO)
	Number 5,000	84($\overset{C=3}{S=1}$, W=XMS*)
	Number 6,000	84($\overset{C=3}{S=1}$, W=SS**)
	Number 7,000	84($\overset{C=3}{S=1}$, W=SBO*)
	Number 8,000	84($\overset{C=3}{S=21}$, W=MNWH)
	Number 9,000	84($\overset{C=3}{S=1}$, W=TSO*)

CARD FORMAT:

	1	6			C						
1	2 3 4 5	6	7 8 9 10	11 12 13 14 15	16 17 18 19 20	21 22 23 24 25	26 27 28 29 30	31 32 33 34 35	36 37 38 39 40	41 42 43 44 45	46 47 48 49 50

CARD TYPE: 17 B_{ba}

SENTENCE CONSTITUENT: 1-99 Number Phrase

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Numbers 1 through 9	11
2	Number 10	None
3	Numbers 11 through 19	13
4	Numbers 20,30,40...90	14
5	Numbers 21 through 29, 31 through 39, 41 through 49, ... 91 through 99	14 11

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
				1	7														C																														

CARD TYPE: 18 B_{bb}

SENTENCE CONSTITUENT: 100-999 Numbers Phrase

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Numbers 100,200...900	15
2	Numbers 101 through 199, 201 through 299, 301 through 399, ... 901 through 999	15 17

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
				1	8														C																														

CARD TYPE: 19 B_a

SENTENCE CONSTITUENT: 1-999 Numbers Phrase

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Numbers 1-99	17
2	Numbers 100-999	18

CARD FORMAT:

				1	9														C																														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

238

CARD TYPE: 20 B_{bc}

SENTENCE CONSTITUENT: 1000-to-9999 Number Phrase

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Numbers 1000,2000,.....9000	16
2	Numbers 1001 through 1999, 2001 through 2999, ... 9001 through 9999	16 19

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
				2	0														C																														

CARD TYPE: 21 ^{B}p

SENTENCE CONSTITUENT: Number Phrase

VARIABLES: M, C

C=	OPTION	ADD CARD TYPES TO LIST:
-	M=0, Number phrase not to be used	None
1	numbers 1 through 999	19
2	numbers 1000 through 9999	20

CARD FORMAT:

				2	1						M								C								N		G																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49

CARD TYPE: 22 Npb

SENTENCE CONSTITUENT: Simple Noun Phrase

VARIABLES: C,N,G,P

C=	OPTION	ADD CARD TYPES TO LIST:
1	The nucleus of the phrase is a non-verbal noun. Option of expressing ownership but not other genetive relationships(symbol 5)	6 4 9 5
2	The nucleus of the phrase is a non-verbal noun with ownership or other genetive relationships expressed by a pronoun	8 4
3	The nucleus of the phrase is a verbal noun emphasizing the door of the deed.	60

CARD FORMAT:

				2	2														C								N		G		P																	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49

CARD TYPE: 23 Npa

SENTENCE CONSTITUENT: Regular Noun Phrase

VARIABLES: K,B,C,Y,N,G,P

C=	OPTION	ADD CARD TYPES TO LIST:
1	Noun phrase has no genitive noun modifiers	21(N=G=0) 22(N=G=P=0) 80(C=2,Y=0)
2	Noun phrase has one or more genitive noun modifiers	21(N=G=0) 85(M=1,K=0,N=G=P=0) 85 22 80(C=2,Y=0)
3	Noun phrase is a number only	21(M=1, N=G=0)
	NOTE: N,G,P apply to nucleus noun of phrase.	

CARD FORMAT:

				2	3										K		B		C			Y					N		G		P																		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 24 N_{pc}

SENTENCE CONSTITUENT: Appositional Noun Phrase

VARIABLES: M,K,B,C,Y

C=	OPTION	ADD CARD TYPES TO LIST:
-	M=0, the phrase is not to be used	None
1	For all cases where M=1	23(N=G=P=0)

CARD FORMAT:

				2	4						M				K		B		C				Y																										
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

Nap

CARD TYPE: 25

SENTENCE CONSTITUENT: Appositional Phrase

VARIABLES: M,C

C=	OPTION	ADD CARD TYPES TO LIST:
-	M=0, phrase is not to be used	None
1	Appositional phrase is a noun or equivalent	24(M=1)
2	Appositional phrase is a prepositional phrase	38(M=1)
3	Appositional phrase is a relative clause	67(D=N=G=P=0)

CARD FORMAT:

2 5 M C

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49

244

CARD TYPE: 26 N_p

SENTENCE CONSTITUENT: General Noun Phrase

VARIABLES: K,B,Y,D,N,G,P

C=	OPTION	ADD CARD TYPES TO LIST:
1	For all cases	23(K=Y=N=G=P=0) 25 (N=G=P=0)

CARD FORMAT:

				2	6										K		B		1				Y		D		N		G		P																		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 27 Dpa

SENTENCE CONSTITUENT: Specific Quantity Adverb Phrase

VARIABLES: K,B,C,Y,N,G

C=	OPTION	ADD CARD TYPES TO LIST:
1	No emphasis placed on the number in the specific quantity	80(M=1,K=0,C=3) 21
2	Some emphasis is placed on the number in the specific quantity. Example: two times.	21(M=1) 80(M=1,K=0,C=3)

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
				2	7										K		B		C				Y																										

CARD TYPE: 28 Dpb

SENTENCE CONSTITUENT: General Quantity Adverb Phrase

VARIABLES: K,B,C

C=	OPTION	ADD CARD TYPES TO LIST:
1	For all cases	80(M=1,K=0,C=4) 80(K=0,Y=0,C=6)

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
				2	8										K		B		1																														

CARD TYPE: 29 Dpc

SENTENCE CONSTITUENT: Qualitative Adverb Phrase

VARIABLES: K,B,Y

C=	OPTION	ADD CARD TYPES TO LIST:
1	For all cases	80(M=1,K=0,C=5) 80(K=0,Y=0,C=6)

CARD FORMAT:

				2	9										K		B		1				Y																										
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 30 Dp

SENTENCE CONSTITUENT: Adverb Phrase

VARIABLES: M,K,B,C,Y

C=	OPTION	ADD CARD TYPES TO LIST:
-	M=0, phrase is not to be used	None
1	Phrase expresses temporal limitation	80(M=1, C=1)
2	Phrase expresses spatial limitation	80(M=1, C=2)
3	Phrase expresses specific quantity	27
4	Phrase expresses general quantity	28
5	Phrase expresses qualitative limitation	29
6	Phrase expresses degree of intensity	80(M=1,K=0,C=6) 80(K=Y=0, C=6) M=0 less intense M=1 more intense
7	Adverbial preposition phrase	38(M=1)

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
				3	0						M				K		B		C				Y																										

CARD TYPE: 31 Ea

SENTENCE CONSTITUENT: Basic Participle Phrase

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	For all cases	None

CARD FORMAT:

				3	1						M								1																								S	←	W	→			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 32* V_b

SENTENCE CONSTITUENT: Verb-Mood Phrase

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	For the special case of the imperative sentence where emphasis is desired (uses infinitive absolute)	NONE ↓

* After coding is completed, remove card from deck

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
				3	2														1																													

CARD TYPE: 33 V_{bb}

SENTENCE CONSTITUENT: Three-Tense Verb Phrase

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Main verb is not present tense copulative	None
2	Main verb is present tense copulative, no emphasis	31(N=0)
3	Main verb is present tense copulative with emphasis	None

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
				3	3														C																A														

CARD TYPE: 34 Vc

SENTENCE CONSTITUENT: Emphatic Verb Phrase

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Emphasis is placed on certainty of verbal action.Example:He will surely live.	33
2	Emphasis is placed on the duration of the verbal action.Example:He continually eats.	
3	No special emphasis is given to verbal action.	

CARD FORMAT:

				3	4														C																													
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49

CARD TYPE: 35* Vaa

SENTENCE CONSTITUENT: Seven-Tense Verb Phrase

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	For all cases	34(A=A)

*NOTE: After coding card is complete, remove card from deck.

CARD FORMAT:

				3	5														1																													
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49

CARD TYPE: 36

Va

SENTENCE CONSTITUENT: Verb Phrase

VARIABLES: C,S,W

C=	OPTION	ADD CARD TYPES TO LIST:
1	For all cases	35(A=A0

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
				3	6														1																A								S	←	W	→			

CARD TYPE: 37 Bc

SENTENCE CONSTITUENT: Definite Number Phrase

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	The number phrase is indefinite	21(M=1)
2	The number phrase is definite:	21(M=1)

CARD FORMAT:

				3	7														C																														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 38 Xp

SENTENCE CONSTITUENT: Prepositional Phrase

VARIABLES: M,K,B,C,Y,R

C=	OPTION	ADD CARD TYPES TO LIST:
-	M=0, phrase is not used	None
1	Object of preposition is a noun	89A(R=R) 26(Y=0)
2	Object of preposition is a pronoun	R=R 1(N=1,K=Y=0) and NOTE 1
3	Object of preposition is a relative clause	89A(R=R) 67(Y=0)
	"R" refers to class of the preposition of the phrase if not already specified	

CARD FORMAT:

				3	8						M				K		B		C				Y										R																
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 39 No

SENTENCE CONSTITUENT: Direct Object Phrase

VARIABLES: K,B,C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Object is a noun	26(K=Y=D=0)
2	Object is a relative clause	67(K=Y=D=0)

CARD FORMAT:

				3	9										K		B		C																														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

Dpd

CARD TYPE: 40

SENTENCE CONSTITUENT: Copulative Adverb Phrase

VARIABLES: K,B,C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Copulative adverb is temporal	80(M=1, C=1, Y=0)
2	Copulative adverb is locative	80(M=1, C=2, Y=0)

CARD FORMAT:

				4	0										K		B		C																														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 41 Rsp

SENTENCE CONSTITUENT: Subject Pronoun Phrase

VARIABLES: K,B,C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Subject is personal pronoun	24(Y=0)*
2	Subject is demonstrative pronoun	

*NOTE: Card 24 is for an optional appositional phrase modifying the subject pronoun. If none, M=0; otherwise, M=1.

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
				4	1										K		B		C																														

CARD TYPE: 42 Nsp

SENTENCE CONSTITUENT: Subject Phrase

VARIABLES: K,B,C,Y,D,N,G,P

C=	OPTION	ADD CARD TYPES TO LIST:
1	Subject is noun	26(Y=D=N=G=P=0)
2	Subject is a pronoun	41
3	Subject is a clause of the general form: (the fact) That	68
4	Subject is an infinitive phrase of the type: to eat an apple.....	56(K=Y=0)

CARD FORMAT:

				4	2										K		B		C				Y		D		N		G		P																		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 43 Nop

SENTENCE CONSTITUENT: Object Phrase		
VARIABLES: K,B,C,Y,D		
C=	OPTION	ADD CARD TYPES TO LIST:
1	Object is indefinite (D=0) (no definite article, or possessive or proper name)	26(Y=D=0)
2	Object is definite or proper name (D=2)	39

CARD FORMAT:

				4	3						M				K		B		C				Y		D																								
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 44 Nip

SENTENCE CONSTITUENT: Indirect Phrase

VARIABLES: K,B,C,Y

C=	OPTION	ADD CARD TYPES TO LIST:
1	Indirect object is noun	38(M=1,Y=0,R=R)
2	Indirect object is infinitive phrase	56(Y=0)

CARD FORMAT:

				4	4										K		B		C				Y										R																
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 45 Npx

SENTENCE CONSTITUENT: Copulative Phrase

VARIABLES: K,B,C,Y,D

C=	OPTION	ADD CARD TYPES TO LIST:
1	Predicate is adjective	4(Y=0)
2	Predicate is adverb	40
3	Predicate is noun	26(Y=D=N=G=P=0)
4	Predicate is pronoun	41
5	Predicate is prepositional phrase	38(M=1, Y=0)
6	Predicate is a circumstantial clause of the type: in order that.....	69(Y=0)

CARD FORMAT:

				4	5										K		B		C				Y		D																								
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

Vma

CARD TYPE: 46

SENTENCE CONSTITUENT: Direct-Object Verb-Modifying Phrase

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Direct object is pronoun (no emphasis)	91(C=3) 30 1 30
2	Direct object is pronoun (emphasis)	30 10 1 30
3	Direct object is noun (or verbal noun)	30 1 43(M=1) 30

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
				4	6														C																													

CARD TYPE: 47 Vmb

SENTENCE CONSTITUENT: Indirect-Object Verb-Modifying Phrase

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Direct object is pronoun (no emphasis) Classical Hebrew	91(C=3) 30 44(R=R) 30
2	Direct object is pronoun (emphasis) Modern Hebrew	30 10 44 (R=R) 30
3	Direct object is noun	30 43 (M=1) 44 (R=R) 30

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
				4	7														C														R															

Vmc

CARD TYPE: 48

SENTENCE CONSTITUENT: Discourse Verb-Modifying Phrase

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Direct object is pronoun(no emphasis) Classical Hebrew	91(C=3) 1 30 72
2	Direct object is pronoun (emphasis) Modern Hebrew	30 10 1 30 72
3	Verb has direct object	30 1 43 (M=1) 30 72
	Verb has no direct object	30 1 43 (M=0) 30 72

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
				4	8														C																														

267

CARD TYPE: 49 Vmd

SENTENCE CONSTITUENT: Double-Accusative Verb-Modifying Phrase

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Direct object is pronoun (no emphasis) Classical Hebrew	91(C=3) 30 1 26 30
2	Direct object is pronoun (emphasis) Modern Hebrew	30 10 1 26 30
3	Direct object is noun	30 1 43(M=1) 26 30

CARD FORMAT:

				4	9														C																														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

Vm

CARD TYPE: 50

SENTENCE CONSTITUENT: Verb-Modifying Phrase

VARIABLES: K,B,C,Y,

C=	OPTION	ADD CARD TYPES TO LIST:
1	Verb is copulative (A=1)	30 45 30
2	Verb is intransitive (A=2)	30
3	Verb is transitive (A=3)	46
4	Verb takes both direct and indirect object (A=4)	47(R=R)
5	Verb is transitive and takes direct object with preposition (A=5)	30 38(M=1,R=R) 30
6	Verb is purpose verb (A=6)	30 56(K=B=Y=0) 30
7	Verb is discourse verb (A=7)	48
8	Verb takes double accusative (A=8)	49

CARD FORMAT:

				5	0										K		B		C				Y										R		A														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 51 Vmr

SENTENCE CONSTITUENT: Objective Relative-Clause Verb-Modifying Phrase

VARIABLES: K,B,C,Y,N,G,P

C=	OPTION	ADD CARD TYPES TO LIST:
1	Verb is copulative (A=1)	30
2	Verb is transitive (A=3)	46 (C=1,2 only)
3	Verb takes both direct and indirect object (A=4)	47(C=1,2 only)
4	Verb is transitive and takes object with a preposition (A=5)	30 1 (M=1, R=R) 30
5	Verb takes a double accusative (A=8)	49(C=1,2 only)

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
				5	1										K		B		C				Y				N		G		P		R		A														

CARD TYPE: 52 Vmi

SENTENCE CONSTITUENT: Indirect Relative-Clause Verb-Modifying Phrase

VARIABLES: K,B,C,Y,N,G,P

C=	OPTION	ADD CARD TYPES TO LIST:
1	Direct object is pronoun (no emphasis) (classical)	91(C=3) 30 1(M=1,R=R, NOTE 1) 30
2	Direct object is pronoun (emphasis)	30 10 1 (M=1,R=R NOTE 1) 30
3	Direct object is noun	30 1(M=1, R=R NOTE 1) 43(M=1) 30

CARD FORMAT:

				5	2										K		B		C				Y										R																
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

Vp

CARD TYPE: 53*

SENTENCE CONSTITUENT: Verb Phrase

VARIABLES: K,B,C,Y,R,A,V,I,T

C=	OPTION	ADD CARD TYPES TO LIST:
1	Verb is copulative (A=1) (R=0)	30 36 (A=1) 50 (C=1,R=0,Y=0)
	Verb is not copulative (A≠1)	30 36 (A=A) 50(C=A,R=R,Y=0)
	*NOTE 1: Remove card from list after coding if specified by governing symbol. NOTE: R,A,V,I,T refer to main verb	

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
				5	3										K		B		1				Y										R		A		V		I		T								

CARD TYPE: 54 Vrb

SENTENCE CONSTITUENT: Objective Relative Clause Verb Phrase

VARIABLES: K,B,C,V,I,T

C=	OPTION	ADD CARD TYPES TO LIST:
1	For all cases	36(A=A) 51(R=R, A=A)

CARD FORMAT:

				5	4										K		B		1														R		A		V		I		T								
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

MS 273

Vri

CARD TYPE: 55

SENTENCE CONSTITUENT: Indirect Relative Clause Verb Phrase

VARIABLES: K,B,C,V,I,T

C=	OPTION	ADD CARD TYPES TO LIST:
1	For all cases	36(A=A) 52(R=R)
NOTE	Grammar presently contains rules for verbs of A=4 only R,A,V,I,T refer to main verb of clause	

CARD FORMAT:

				5	5										K		B		1														R		A		V		I		T								
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 56 Nv

SENTENCE CONSTITUENT: Infinitive Construct Phrase

VARIABLES: K,B,C,Y,R,A

C=	OPTION	ADD CARD TYPES TO LIST:
1	Subject of action not named or previously named in-sentence	96(C=A) 50(C=A,R=R)
2	Subject of action is pronoun (no emphasis)	96(C=A) 91(C=3) 50(C=A,R=R)
3	Subject is noun or emphasized pronoun	96(C=A) 42 50(C=A,R=R)
	R and A refer to attributes of infinitive	

CARD FORMAT:

				5	6										K		B		C				Y										R		A														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 57 Nw

SENTENCE CONSTITUENT: Infinitive Absolute Phrase

VARIABLES: K,B,C,Y,R,A

C=	OPTION	ADD CARD TYPES TO LIST:
1	For all cases	95 50(K=Y=N=G=P=0)

CARD FORMAT:

				5	7										K		B		1				Y										R		A														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

Epb

CARD TYPE: 58

SENTENCE CONSTITUENT: Construct Participle Phrase

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Object is noun	82 (C=3) 23
2	Object is pronoun	82 91 (C=3)

CARD FORMAT:

				5	8														C																													
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49

CARD TYPE: 59 Epa

SENTENCE CONSTITUENT: Absolute/Construct Participle Phrase

VARIABLES: C,R,A,V

C=	OPTION	ADD CARD TYPES TO LIST:
1	The phrase is of the form: (the one who) eats apples	31 50
2	The phrase is of the form: the eater of apples A=3, V=1	58

CARD FORMAT:

				5	9														C														R		A		V												
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 60*

SENTENCE CONSTITUENT: Participle Phrase

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	For all cases	59

*NOTE: After this card is coded, remove card from deck.

CARD FORMAT:

				6	0														1																														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 61

SENTENCE CONSTITUENT: Possessive Independent Clause

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Emphasis on possessor	38(M=1,Y=0,R=1) 36(A=1,S=12,W=HYH*) 42(N=G=P=0) 30
2	Emphasis on thing possessed	42(N=G=P=0) 36(A=1,S=12,W=HYH*) 38(M=1,Y=0,R=1) 30
3	No emphasis	36(A=1,S=12,W=HYH*) 38(M=1,Y=0,R=1) 42(N=G=P=0) 30
4	Special case for Present Tense, active voice, indicative mood. non-negated possession	93(M=1,C=3) 38(M=1,Y=0,R=1) 42(N=G=P=0) 30
	Negated possession, present tense, active voice, indicative mood.	38(M=1,Y=0,R=1) 42(N=G=P=0) 30
NOTE:	38 defines possessor 42 defines thing possessed	

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
				6	1														C																														

Sab

CARD TYPE: 62

SENTENCE CONSTITUENT: Definite Independent Clause

VARIABLES: C

C=	OPTION		ADD CARD TYPES TO LIST:
1	General Case		42(N=G=P=0) 53(Y=V=I=T=0)
	Special cases for present tense, active voice, indicative mood.		
2	Subject is noun, verb is positive copulative	Classical with emphasis	93(M=1,C=3) 42(K=Y=N=G=P=0,C=1) 53(Y=V=I T=0)
		Modern or classical with no emphasis	93(M=0,C=3) 42(K=Y=N=G=P=0,C=1) 53(Y=V=I=T=0)
	Subject is Pronoun verb is positive copulative. no emphasis		93(M=0,C=3) 42(K=Y=N=G=P=0,C=2) 53(Y=V=I=T=0)
3	Subject is Pronoun (with emphasis) Verb is any non-negated verbal (Otherwise use class 1)		53(Y=V=I=T=0)
4	Subject is Pronoun and verb is any negated verbal		53(Y=V=I=T=0)
5	Subject is Noun and verb is any negated verbal (no emphasis)		26(K=Y=N=G=P=0) 53(Y=V=I=T=0)
6	Same as 5 with emphasis on verbal idea		Same as 5

CARD FORMAT:

				6	2														C																													
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49

CARD TYPE: 63* Sac

SENTENCE CONSTITUENT: Indefinite Independent Clause

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	For all cases, however, ambiguity is produced by present tense. Avoid that tense here.	53(K=Y=V=I=T=0)

* NOTE: After coding this card, remove from list.

CARD FORMAT:

				6	3														1																														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 64 Sa

SENTENCE CONSTITUENT: Independent Clause

VARIABLES: K,B,C,N,G,P,V,I,T

C=	OPTION	ADD CARD TYPES TO LIST:
1	Clause expresses possession (subject is thing possessed)	61
2	Subject is not named (no emphasis or previously named in sentence)	63
3	Subject is named by either noun or pronoun (or equivalent)	62
	NOTE: N,G, and P refer to subject of clause V,I,T, refer to verb of clause for compound subject: N=1 for disjunctive string of singulars otherwise N=3 G=2 for string of feminines otherwise G=1 P=1 if any subject element is 1st person =2 if any subject element is 2nd person and no 1st person otherwise P=3	

CARD FORMAT:

				6	4										K		B		C								N		G		P						V		I		T								
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 65 Sro

SENTENCE CONSTITUENT: Objective Relative Phrase

VARIABLES: C,Y,N,G,P

C=	OPTION	ADD CARD TYPES TO LIST:
1	Subject of clause is not named, or previously named in sentence	54
2	Subject of clause is noun, pronoun for emphasis or equivalent	42(N=G=P=0) 54

CARD FORMAT:

				6	5														C				Y				N		G		P																		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 66 Sri

SENTENCE CONSTITUENT: Indirect Relative Clause

VARIABLES: C,Y,N,G,P

C=	OPTION	ADD CARD TYPES TO LIST:
1	Subject of clause is not named	55
2	Subject of clause is a noun, pronoun, or equivalent	42(N=G=P=0) 55

CARD FORMAT:

				6	6														C						Y		N		G		P																		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 67 Rg

SENTENCE CONSTITUENT: Relative Pronoun Clause

VARIABLES: K,B,C,Y,D,N,G,P

C=	OPTION	ADD CARD TYPES TO LIST:
1	The relative pronoun is subject of clause	53
2	The relative pronoun is object of the verb	65
3	The relative pronoun is indirect object of the verb	66

CARD FORMAT:

				6	7										K		B		C				Y		D		N		G	P																			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 68 Kn

SENTENCE CONSTITUENT: Subject-Object Dependent Clause

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Clause is introduced by KY (Biblical)	64
2	Clause is introduced by S (Modern)	64

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
				6	8														C																													

CARD TYPE: 69 Kc

SENTENCE CONSTITUENT: Circumstantial Dependent Clause

VARIABLES: K,B,C,Y

C=	OPTION	ADD CARD TYPES TO LIST:
1	Circumstance is time dependent: "when...."	79(C=4) 64
2	Circumstance is purpose-result dependent: "in order that...."	79(C=5) 64(T=2 or 5)
3	Circumstance is cause-reason dependent: "because ......"	79(C=6) 64
4	Accompanying circumstance expressed by a prepositional phrase: "in the morning....."	38(M=1, Y=0)

CARD FORMAT:

				6	9										K		B		C				Y																										
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 70 K_k

SENTENCE CONSTITUENT: Conditional Clause

VARIABLES: K,B,C,Y

C_α	OPTION	ADD CARD TYPES TO LIST:
1	All cases	64(V=I=T=0)

CARD FORMAT:

				7	0										K		B		1				Y																										
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

Ki

CARD TYPE: 71

SENTENCE CONSTITUENT: Interogative Clause

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Truth of sentence is questioned	90 64(N=G=P=0)
2	Subject of sentence is questioned	91(C=5) 53(Y=0)
3	Object of verb is questioned	91(C=5) 7(Y=N=G=P=0, R=R)
4	Indirect object of verb is questioned	89A 91(C=5) 66(Y=N=G=P=0)
	NOTE: N,G,P refer to subject of clause R refers to verb of clause	

CARD FORMAT:

				7	1														C								N		G		P		R																
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

Kd

CARD TYPE: 72

SENTENCE CONSTITUENT: Discourse Clause

VARIABLES: K,B,C,Y

C=	OPTION	ADD CARD TYPES TO LIST:
1	Indirect discourse	68
2	Direct discourse	76(Y=0)

CARD FORMAT:

				7	2										K		B		C			Y																											
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

Sd

CARD TYPE: 73

SENTENCE CONSTITUENT: Dependent Clause Sentence

VARIABLES: C

C=	OPTION	ADD CARD TYPES TO LIST:
1	Emphasis on dependency	69 92(C=3)* 64(V=I=T=0)
2	No emphasis on dependency	64(V=I=T=0) 92(C=3)* 69
	* Provision for an optional comma	

CARD FORMAT:

				7	3														C																														
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

S

CARD TYPE: 74

SENTENCE CONSTITUENT: Basic Sentence

VARIABLES: C,I,T

C=	OPTION	ADD CARD TYPES TO LIST:
1	No dependency or conditions are expressed	64(I=T=0)
2	Dependency clause or phrase is expressed	73
3	Conditions are expressed *	70 92(C=3) 64(V=I=T=0)
	*NOTE: Tense of Protasis(70) and Apodasis (64) is governed by tense of sentence (76). Negation of Apodosis (64) is governed by sentence (76). Negation of Protasis (70) is an independent variable to be specified.	

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
				7	4														C																				I		T							

CARD TYPE: 75 Si

SENTENCE CONSTITUENT: Interogative Sentence

VARIABLES: K,B,C

C=	OPTION	ADD CARD TYPES TO LIST:
1	No dependency or condition expressed	71
2	Dependency is expressed, no emphasis	69 71
3	Dependency is expressed with emphasis	71 69

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
				7	5										K		B		C																														

CARD TYPE: 76 Sc

SENTENCE CONSTITUENT: Completed Sentence

VARIABLES: K,B,C,Y

C=	OPTION	ADD CARD TYPES TO LIST:
1	Declarative sentence	79(C=8) 74
2	Interogative sentence	79(C=8) 75
3	Imperative sentence (emphatic using!)	79(C=8) 74(I=T=2)
	(NOTE: Declarative sentence here includes indicative, subjunctives and non-emphatic imperatives.) Symbol 79 is an optional logical connective like "but,..."	

CARD FORMAT:

				7	6						1				K		B		C				Y																										
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 77

SENTENCE CONSTITUENT: Adjective A

VARIABLES: C,S,W

C=	OPTION	ADD CARD TYPES TO LIST:
1	For all cases	None
NOTE:	Specify S and W from dictionary.	

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
				7	7														1																								S	←	W	→			

296

CARD TYPE: 78A*

SENTENCE CONSTITUENT:		
VARIABLES:		
C=	OPTION	ADD CARD TYPES TO LIST:
1		78(C=1)
2		78(C=2)

*NOTE: Remove card after coding

CARD FORMAT:

				7	8														C																													
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49

B

CARD TYPE: 78

SENTENCE CONSTITUENT: Number Absolute

VARIABLES: C,S,W,H

C=	OPTION	ADD CARD TYPES TO LIST:
1	Number 1(S=39, W=AXD*, H=1)	NONE
2	Number 2(S=30, W=SN**, H=0)	
3	Numbers 3 through 9, S and W are specified from governing card type	
4	The number 10(S=1, W=OSR*, H=0)	
5	The number 100(S=1, W=MA**, H=0)	
6	The number 1000(S=1, W=ALP*, H=0)	

CARD FORMAT:

				7	8														C																								S	←	W	→			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

H
52

CARD TYPE: 79

SENTENCE CONSTITUENT: Conjunctives C

VARIABLES: C,S,W

C=	OPTION	ADD CARD TYPES TO LIST:
1	"and" (S=1, W=W***)	NONE
2	"or" (S=1, W=AW**)	
3	"that" (S=1, W=KY**)	
4	Temporal conjunctives: "When, before, until, after..."	
5	Purpose conjunctives: "in order that.."	
6	Cause conjunctives: "because,..."	
7	Discourse conjunctives: "saying,..."	
8	Logical connectives "but,..."	
	NOTE: Specify S and W from dictionary when not given.	

CARD FORMAT:

				7	9						M								C																								S	←	W	→		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49

CARD TYPE: 80

SENTENCE CONSTITUENT: Adverb D

VARIABLES:

C=	OPTION	ADD CARD TYPES TO LIST:
-	M=0, adverb not used	NONE ↓
1	Temporal adverbs	
2	Locative adverbs	
3	Qualitative Adverbs	
4	Adverbs of Intensity	
5	Pluperfect adverb (S=1, W= MS**)	
6	Future perfect adverb (S=1, W=LKS*)	
Note:	Specify S and W from dictionary.	

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
				8	0						M				K		B		C				Y																				S ←		W →				

CARD TYPE: 81

SENTENCE CONSTITUENT:	Participles Absolute	E
VARIABLES:	A,K,B,C,V,S,W	

A=	OPTION	ADD CARD TYPES TO LIST:
		NONE ↓
1	Copulative verbal (A=1)	
2	Intransitive verbal (A=2)	
3	Transitive verbal (A=3)	
4	Indirect object verbal (A=4)	
5	Prepositional verbals (A=5)	
6	Purpose verbals (A=6)	
7	Discourse verbals (A=7)	
8	Double accusative verbals (A=8)	
NOTE:	Specify S and W from dictionary.	

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
				8	1										K		B		1																A		V						S	←	W		→		

CARD TYPE: 82

SENTENCE CONSTITUENT: Participles Construct		G
VARIABLES: K,B,C,A,V,S,W		
A=	OPTION	ADD CARD TYPES TO LIST:
1	Copulative verbal (A=1)	NONE ↓
2	Intransitive verbal (A=2)	
3	Transitive verbal (A=3)	
4	Indirect object verbal (A=4)	
5	Prepositional verbals (A=5)	
6	Purpose verbals (A=6)	
7	Discourse verbals (A=7)	
8	Double accusative (A=8)	
NOTE:	Specify S and W from dictionary.	

CARD FORMAT:

				8	2										K		B		1																A		✓						S	←	W		→		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 83

SENTENCE CONSTITUENT: Definite Article H

VARIABLES: C,S,W

C=	OPTION	ADD CARD TYPES TO LIST:
1	For all cases (S=1,W=H***)	NONE

CARD FORMAT:

				8	3														1																								1	H	*	*	*		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 84A*

SENTENCE CONSTITUENT:

VARIABLES:

C=	OPTION	ADD CARD TYPES TO LIST:
1		84(C=1)
2		84(C=1)

* NOTE: Remove card after coding

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
				8	4														C																														

CARD TYPE: 84

SENTENCE CONSTITUENT: Number Construct I

VARIABLES: C,S,W

C=	OPTION	ADD CARD TYPES TO LIST:
1	Number 1 (S=39, W=AXD*)	NONE ↓
2	Number 2 (S=30, W=SN**)	
3	Numbers 3 through 9, S and W are specified from governing card type	
4	The number 10 (S=1, W=OSR*)	
5	The number 100 (S=1, W=MA**)	
6	The number 1000 (S=1, W=ALP*)	

CARD FORMAT:

				8	4														C																								S ←		W →				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 85

SENTENCE CONSTITUENT: Nouns Construct J

VARIABLES: M,K,B,N,G,P,S,W

C=	OPTION	ADD CARD TYPES TO LIST:
-	M=0, noun is not to be used	NONE
1	For all cases -- no proper names.	
NOTE:	Specify S and W from dictionary	

CARD FORMAT:

				8	5						M				K		B		1								N		G		P												S ←		W →				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

H X
52 54

CARD TYPE: 86

SENTENCE CONSTITUENT: Negatives L

VARIABLES: L,S,W

C=	OPTION	ADD CARD TYPES TO LIST:
L=1	S=1, W=LWA*	NONE
L=2	S=1, W=AYN*	↓
L=3	S=33, W=LBLT	
L=4	S=1, W=AL**	

CARD FORMAT:

				8	6														1		L																						S ←		W →				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 87

SENTENCE CONSTITUENT: Noun Absolute N

VARIABLES: C,S,W

C=	OPTION	ADD CARD TYPES TO LIST:
1	Non-personal nouns	NONE
2	Proper nouns	
3	Non-Hebrew Words	Add word to Non-Hebrew work list at extreme end of syntax cards.
NOTE:	Specify S and W from dictionary	

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
				8	7										K		B		C																								S		W				

H
52

X
54.

CARD TYPE: 88

SENTENCE CONSTITUENT: Object Particle O

VARIABLES: C,S,W

C=	OPTION	ADD CARD TYPES TO LIST:
1	S=1, W=AWT*	NONE ↓

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
				8	8														1																								1						

CARD TYPE: 89A*

SENTENCE CONSTITUENT:		
VARIABLES:		
	OPTION	ADD CARD TYPES TO LIST:
R=1,7		NONE
R=8		89(R=R)

*Remove card from list after coding

CARD FORMAT:

				8	9														C														R																
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 89*

SENTENCE CONSTITUENT: Preposition P

VARIABLES: S,W

C=	OPTION	ADD CARD TYPES TO LIST:
R=1	"to" (S=1,W=L***)	NONE
R=2	"in" (S=1,W=B***)	
R=3	"like/as" (S=1,W=K***)	
R=4	"from" non-emphatic (S=1,W=M***)	
R=5	"from" emphatic (S=1, W=MN**)	
R=6	"to" (S=1,W=AL**)	
R=7	"on" (S=1,W=OL**)	
R=8	others - Specify S and W from dictionary	

*Remove from list if R < 8

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
				8	9														1														R										S		W			

CARD TYPE: 90

SENTENCE CONSTITUENT: Interogative Q

VARIABLES: C,S,W

C=	OPTION	ADD CARD TYPES TO LIST:
1	Special case. Questions truth of sentence in which subject is not named. (Biblical) Otherwise use 2,	. NONE
2	Questions truth of a sentence.	
3	Questions time, manner, purpose, etc. of a statement. Specify S and W from dictionary.	

CARD FORMAT:

				9	0														C																								S	←	W	→			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 91

SENTENCE CONSTITUENT: Pronoun R

VARIABLES: C,N,G,P,S,W

C=	OPTION	ADD CARD TYPES TO LIST:
1	Demonstratives (S=W=0)	NONE
2	Subject personal pronouns (S=W=0)	
3	Object personal pronouns (S=W=0)	
4	Relative pronoun (N=G=P=S=W=0)	
5	Interogative pronouns*	
*NOTE:	Specify S and W from dictionary	

CARD FORMAT:

				9	1														C								N		G		P												S	←	W	→			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50

CARD TYPE: 92

SENTENCE CONSTITUENT: Punctuation Mark T

VARIABLES: M,C

C=	OPTION	ADD CARD TYPES TO LIST:
-	M=0 punctuation to be omitted	NONE
1	"left quote	
2	"right quote	
3	, comma	
4	:colon	
5	? question mark	
6	. period	
7	! exclamation mark	
8	- maqqeph	

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
				9	2						M								C																														

CARD TYPE: 93

SENTENCE CONSTITUENT: Particle U

VARIABLES: M,C,S,W

C=	OPTION	ADD CARD TYPES TO LIST:
-	M=0, particle to be omitted	NONE
1	Possessive (S=1,W=SL**)	
2	Subjunctive (S=1,W=NA**)	
3	Copulative (S=1,W=YS**)	
4	Conditional particle, past tense (S=1, W=LWLY)	
5	Negative conditional (S=1,W=LWLY)	
6	Conditional particle, future tense (S=1,W=AM**)	
7	Locative particle (S=1,W=H***)	

CARD FORMAT:

	9	3		M	C					S←	W →
1	2 3 4 5	6	7 8 9 10	11 12 13 14 15	16 17 18 19 20	21 22 23 24 25	26 27 28 29 30	31 32 33 34 35	36 37 38 39 40	41 42 43 44 45	46 47 48 49 50

CARD TYPE: 94

SENTENCE CONSTITUENT: Verb V

VARIABLES: C,S,W

ADD CARD TYPES TO LIST: NONE

A=	OPTION
1	Copulative (S=12,W=HYH*)
2	Intransitive verbs
3	Transitive verbs
4	Indirect object verbs
5	Prepositional verbs
6	Purpose verbs
7	Discourse verbs
8	Double accusative verbs
NOTE:	Specify S and W from dictionary

CARD FORMAT:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50
				9	4														1																A								S ←		W →				

CARD TYPE: 95

SENTENCE CONSTITUENT: Infinitive Absolute W

VARIABLES: C,S,W

C=	OPTION	ADD CARD TYPES TO LIST:
1	Copulative (S=11,W=HYH*)	NONE
2	Intransitive verbs	
3	Transitive verbs	
4	Indirect object verbs	
5	Prepositional verbs	
6	Purpose verbs	
7	Discourse verbs	
8	Double accusative berbs	
NOTE:	Specify S and W from dictionary	

CARD FORMAT:

				9	5														1																A									S←	W→			
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49

CARD TYPE: 96

SENTENCE CONSTITUENT: Infinitive Construct Y

VARIABLES: C,S,W

C=	OPTION	ADD CARD TYPES TO LIST:
1	Copulative (S=11,W=HYH*)	NONE
2	Intransitive verbs	
3	Transitive verbs	
4	Indirect object verbs	
5	Prepositional verbs	
6	Prupose verbs	
7	Discourse verbs	
8	Double accusative verbs	
NOTE:	Specify S and W from dictionary	

CARD FORMAT:

				9	6														1																A								S ←		W →				
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50